Linux based 3G Multimedia Mobile-phone Application Programming Guide [Telephony Service]

Draft 1.0

NEC Corporation
Panasonic Mobile Communication Ltd.

Contents

1 The Overview of MPP Application Programming Guide

1.1 Overview

This document describes a programming guideline for developers who develop applications using API standardized by Mobile Phone Profile Working Group of CE Linux Forum

This document explains how to use API or programming model to develop applications.

The Outlines explained in each paragraph are described below.

Chapter 2
This paragraph describes the programming model. Firstly, how to use API and the common use for API are explained. Second, how to use the component (MSB: Mobile Software Bus) which supports communications between applications and between middleware components.

Chapter 3
This paragraph describes how to use API with an example of Telephony Services API and explains how to use API referring to Sequence example for Circuit Switched Communication Service, Packet Switched Communication Service, Short Message Service, and Equipment Service.

In addition, for the detail of each API stated in this document, please refer to Linux based 3G Multimedia Mobile-phone API Specification which is CE Linux Forum Technical Document.

1.2 Terminology/Abbreviation

A-CPU:	Application CPU
App :	Application programs
API :	Application Program Interface
ASF :	Advanced Streaming Format
C-CPU:	Communication CPU
CDF :	Certificate Download Function
CS :	Circuit Switched network
Engine :	Application Engine
Elib :	Extended Library
FLASH :	Macromedia Flash Player
glibc :	GNU C Library
GTK+ :	The GIMP Toolkit plus
HTTP :	Hyper Text Transfer Protocol
I/F :	Interface
IMEI :	International Mobile Equipment Identity
IR :	Infra-Red
JAM :	Java Application Manager
KVM :	K Virtual Machine
Kernel :	Linux Kernel
LCD :	Liquid Crystal Display
MAW :	Monitoring and Watching
MSB :	Mobile Software Bus
OBEX :	Object Exchange protocol
OCR :	Optical Character Reader
PIN :	Personal Identification Number
PS :	Packet Switched network
SCA :	Stream Control API
SDFS :	SD File System
SMS :	Short Message Service
SSL :	Secure Socket Layer
TAF :	Terminal Adaptation Function
WDC :	Watching Device Condition
X11R6 :	X Window System Release 6

2 Programming Model

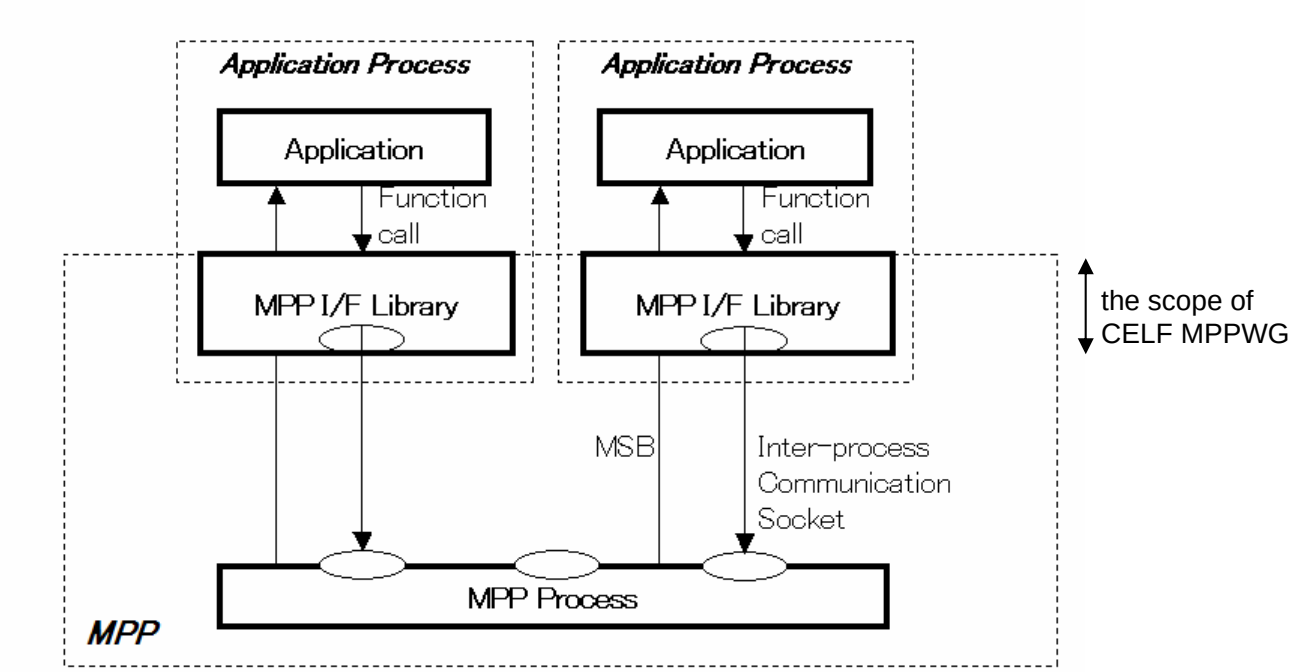
2.1 Programming Model(common to all)

2.1.1 How to use API (common to all)

2.1.1.1 Communication between Application and MPP blocks

MPP provides Library (MPP I/F Library) and function calls are used from application to MPP I/F library. Inter-process communication sockets are used from MPP I/F library to MPP processes.

For the notification from MPP to Application, notification by primitive is used through MSB.



Application process composes application program and MPP I/F library.

Application process:

is process of Linux and composes 1 application by 1 process. Applications hold values which discriminated whole by application ID system.

MPP process：

The function provides MPP realizes Linux process or library. 1 process provides application program over 1 common library.

When application program requests job for MPP, application program starts MPP I/F library by using function call of C description. Moreover, MPP I/F library starts MPP process Linux using socket function which is Inter-process communication.

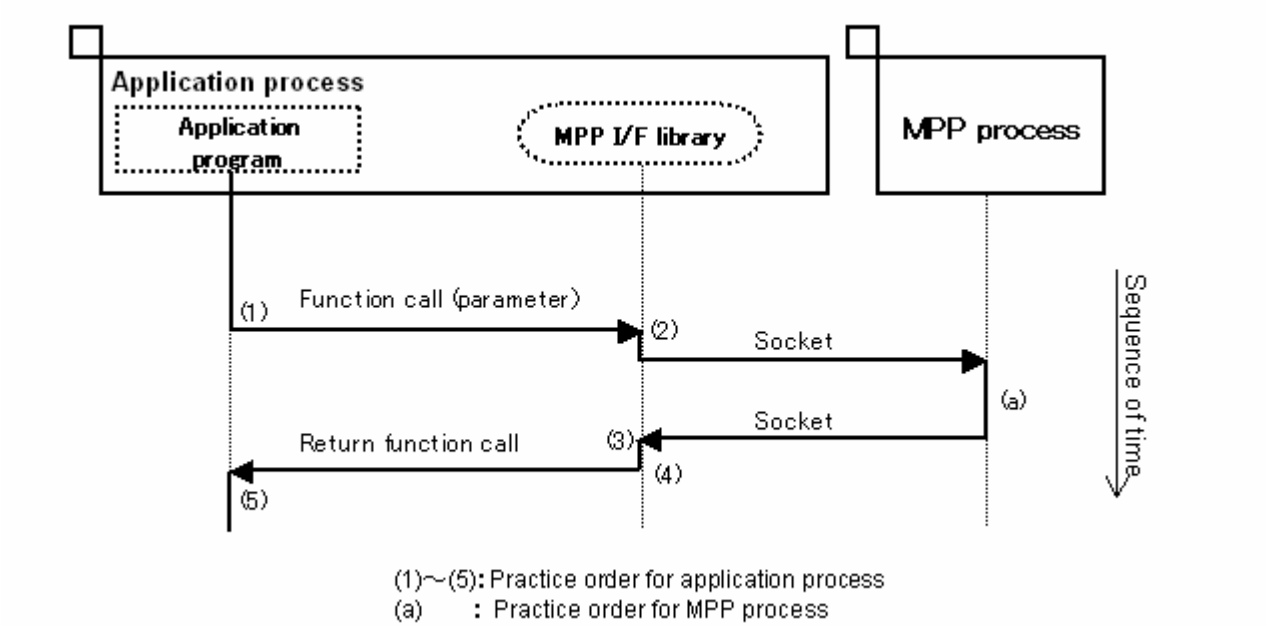
When notificates the event occurred from MPP library to application program and with data, it notificates to MSB.The event occurred means the following: application program for hardware could not expect or the job which application program requests MPP closed.

The function which describes by API Specification is an interface which are application program and the function call between MPP I/F library. MPP I/F library is congregation these functions.

2.1.1.2 API for MPP I/F Library

There are two types of APIs of MPP I/F libraries provided by MPP; synchronous type and asynchronous type.

(1) Synchronous type

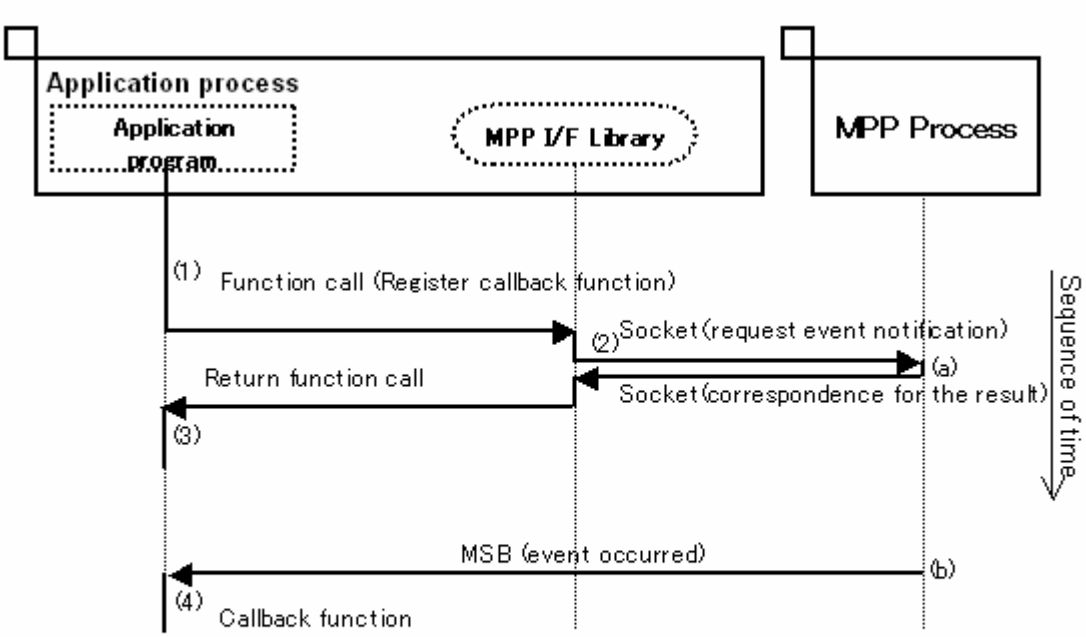


(1) An application program sets up an argument and calls a function of MPP I/F library.
(2) The function of MMP I/F library communicate with MPP process with socket communication and the data is housed in a shared memory.
(3) Receive a notice from MPP process by the socket and the data is housed in the shared house.
(4) Operate processes by following the notice received from MPP process and return a control to the application process. Data to be noticed to the application program is housed in the address pointed out by an argument pointer or the return value of a function.
(5) Operate processes using noticed data.

(a) MPP process operates processes after receiving a notice from the application process by the socket. MPP process receives data from the application process using the shared memory. When the process is finished, it notices that the process is finished, with the socket. MPP process sends data to the application process, using the shared memory.

Shared memory:
A structure that several processes share a memory zone. The shared memory structure makes the memory zone where the processes can read and write.

(2) Asynchronous type



(1)～(4): Practice order for application process
(a)～(b) : Practice order for MPP process

(1) The application program calls a function to be noticed to MPP I/F with [1] A phenomenon to receive or [2]A function name which is to be processed in the case that the phenomenon occurs, as an argument.

[1] : The collection of phenomenon that the application program should treat is defined and in the elements, the application program can point out which phenomenon to revive.

[2] : The function to be processed in the case that the phenomenon occurred is called 'callback function'. Call back function is operated in the context same as the application program.

(2) MPP I/F library communicates with MPP process with the socket to notice the phenomenon occurred. MPP I/F library receive a response that MPP process started to observe when the data of [1]and[2] are noticed to MPP process.

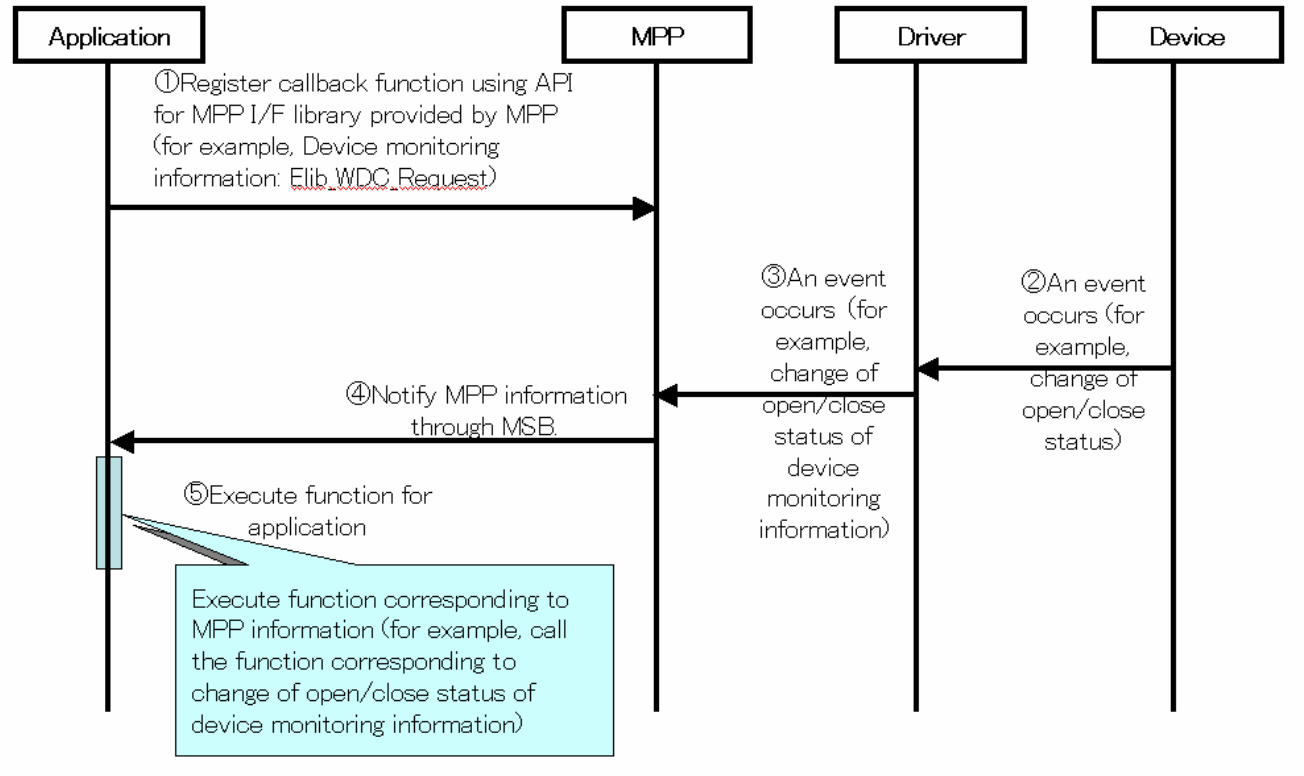
(3) The application process keeps to operate processes.

(a) MPP process records data of [1]and[2] and starts to observe an occurrence of phenomenon to receive.

(b) When the phenomenon occurred, MPP process judges whether to notice or not and in the case to notice, notice to the application program with MSB. MPP process notices detailed data decided by the phenomenon occurred to the application program.(The relation between the phenomenon and detailed data is stated in an applicable function of API Specification.) This data is housed in the sheared library and the address is passed to the call back function as an argument.

(4) A control is passed to the call back function by following OS scheduling and operates processes basing detailed data decided by the phenomenon from MPP process.

When an even occurs, message from MPP can be received by registering callback function at application side using API for MPP I/F library provided by MPP.



Application needs to call Elib_WDC_Request provided by MPP device service to initialize event notification request. As the arguments of Elib_WDC_Request, the type of event to be occurred (in this case the event of change between open/close status) and function to be called back when the event occurs.
When the open/close status of mobile phone is changed, event notification is taken place from MPP to application through MSB, and pre-registered callback function is called.

When the message notification from MPP becomes unnecessary, the function for an event notification cancel request (in this case Elib_WDC_Cancel) is called.

2.1.2 Development Guideline for Application Framework MSB Applications

2.1.2.1 Overview

MSB is a component that supports communication between applications and middleware components.
MSB supports the inter-component communication above sockets. Applications can communicate with each other by using MSB without considering the lower-level communication system. Add in examples for GUI Toolkits that are compatible with MSB. Each application can have a GUI event, a key event, and a communication event at the same time. Fig.1-1 shows the concept of inter-component communication using MSB.

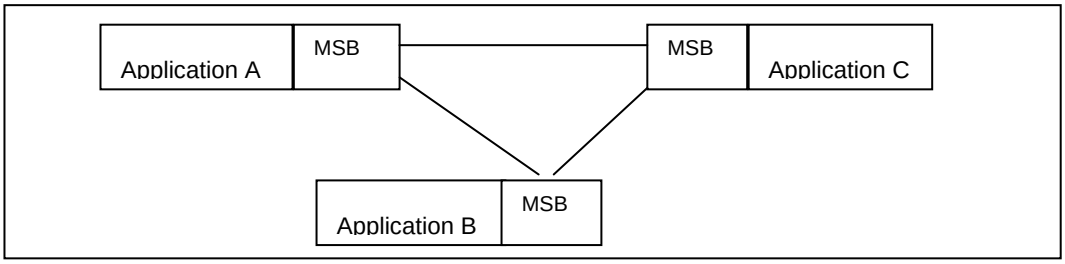


Fig.1-1 Conceptual diagram of inter-component communication after MSB installation

(1) MSB Communication Objects

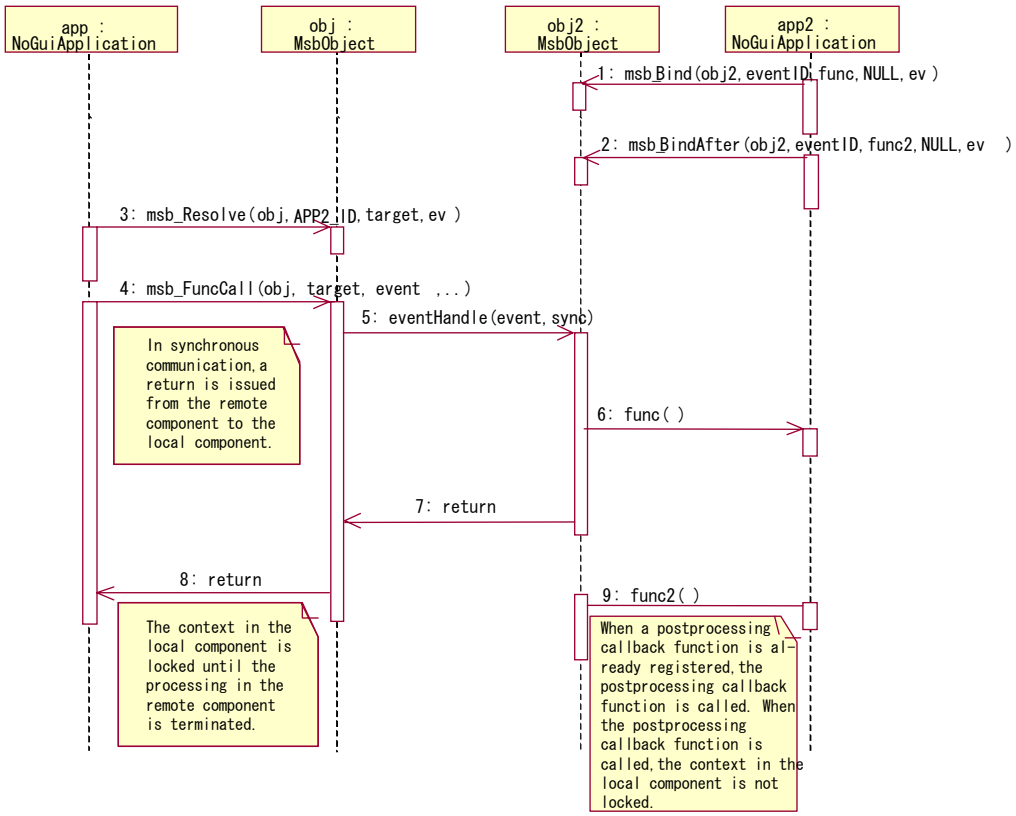
For communication using MSB, each MSB component must have a unique object ID. To start communication, a component acquires a communication object by using the remote-component object ID as a key. By using the acquired communication object as a remote component, the MSB-using communication is performed.
A communication event must be defined for each component by using an event ID.
Object ID and event ID must be uniquely identified in the system.

2.1.2.2 Communication Sequence Using MSB

MSB supplies the function by which events defined by objects are transferred between objects.

(1) Synchronous Communication Sequence

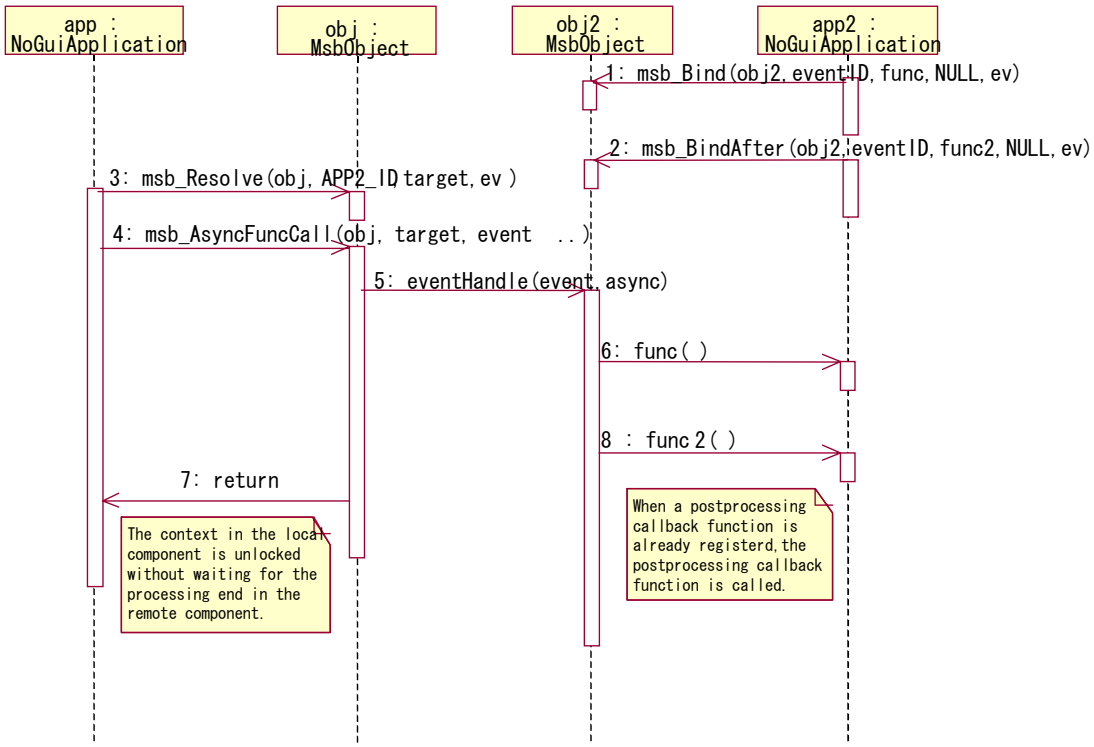
Using the synchronous communication function supplied by MSB, communication between a local component (sending component) and a remote component (destination component) is processed like a library call. As compared with synchronization between components in asynchronous communication, communication from a local component to a remote component is regarded as a library call, and communication status transition of the local component is reduced. However, the operation of the local component is locked until a return occurs in response to the communication event that was posted from the local component to the remote component. The developer must carefully consider this condition. If event processing by the remote component takes a too long time, asynchronous communication should be used and synchronization between the components must be taken.



1. Register callback function "func()" in obj2.
2. Register post-processing callback function "func2()" in obj2.
3. Resolve the name by using "APP2_ID" (object ID) as a key, and acquire the obj2 that corresponds to app2.
4. Send an event to obj2 in object synchronous communication.
5. Send the event to obj2.
6. Call the "func()" (callback function) that corresponds to the event. Store the results in the event.
7. Return the event (results).
8. Return the event (results).
9. Call the "func2()" (postprocessing callback) that corresponds to the event.

(2) Asynchronous Communication Sequence

In asynchronous communication, control returns to the local object without waiting for the return of the communication event sent from the local component (sending component) to the remote component (destination component). Therefore, asynchronous communication is used for event sending without return or used for asynchronous.
If event processing in communication components takes a too long time, use the asynchronous communication.
The following figure shows the sequence in asynchronous communication.



1. Register callback function "func()" in obj2.
2. Register postprocessing callback function "func2()" in obj2.
3. Resolve the name by using "PP2_ID" (object ID) as a key, and acquire the obj2 that corresponds to app2.
4. Sends an event to obj2 in object asynchronous communication.
5. Send the event to obj2.
6. Call the "func()" (callback function) that corresponds to the event. Store the results in the event.
7. Return the success results of the object asynchronous communication.
8. Call the "func2()" (postprocessing callback) that corresponds to the event.

2.1.2.3 Callback Function

The receiving-side processing for inter-object communication using MSB can be registered as a callback function. Register the event ID (which is called from the sending side), and register the function to be called (this is named a callback function). When necessary, a postprocessing callback function to be used after the above callback function is executed can be registered at the receiving side. (That is, one postprocessing callback function can be assigned to one normal callback function.)

2.1.2.4 Object ID

An object ID is a generic name of identifiers of MsbObject in the system. An object ID includes a class ID and identifiers in the same class.
A class ID (MsbObjectID) is the identifier required to use more than one MsbObject having the same function. An identifier in the same class is specified with two low-order columns of MsbObjectID, and is used to identify the MsbObject having the same class ID.

2.1.2.5 Event ID

An event ID (MsbEventID) is the identifier of an event to be processed by MsbObject. Event IDs must be unique in the system.

3. How to use API

3.1 How to user Telephony API

3.1.1 Circuit Switched Communication Service

3.1.1.1 Voice Communication

Functional explanation
'Voice communication' is the service for two-way interactive or one-way active voice communications.

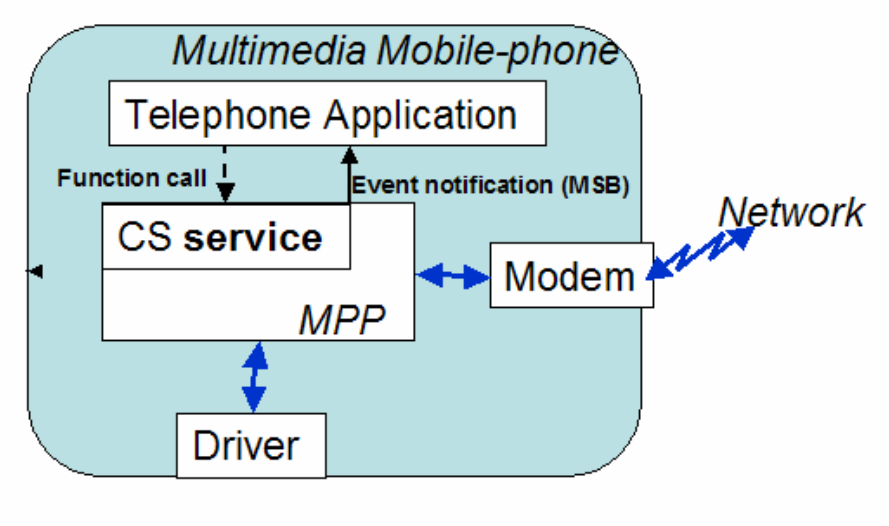
Function provided by MPP
MPP provides following functions to develop an application (called telephone AP as below) which is able to execute basic telephony operations.

- Provide functions such as 'send', 'receive' or 'cut off' of the voice communication.
- Display 'Communicating' Graphical User Interface.
- etc...

Functional classification

Item #	functionality	Functional overview and examples
1	Dialing	・ Perform dialing operation with voice communication from mobile phone. ・ Perform generation of a dialing screen at communication status and response to user operation, etc.
2	Receiving incoming call	・ Receive incoming call from voice communication. ・ Perform generation of a dialing screen at communication status and response to user operation, etc.
3	Communication	・ Sending/receiving voice data between driver and modem. ・ Perform displaying screen during voice communication and mobile phone operation during displaying screen
4	Disconnection	・ Perform disconnection operation of voice communication from mobile phone. ・ Perform disconnection operation of voice communication. ・ Perform displaying screen at disconnection and response to user operation, etc..

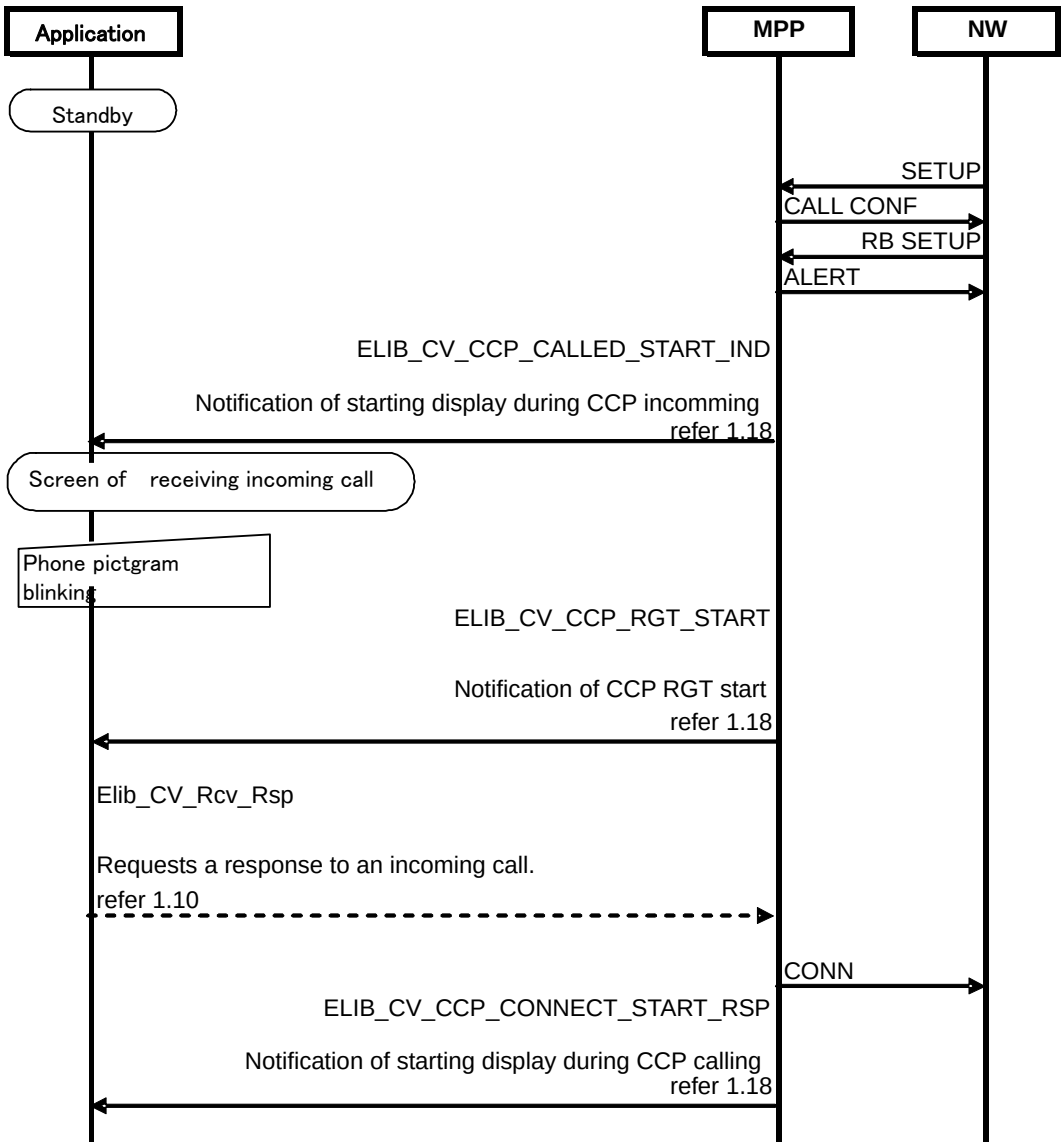
(1) Block Diagram

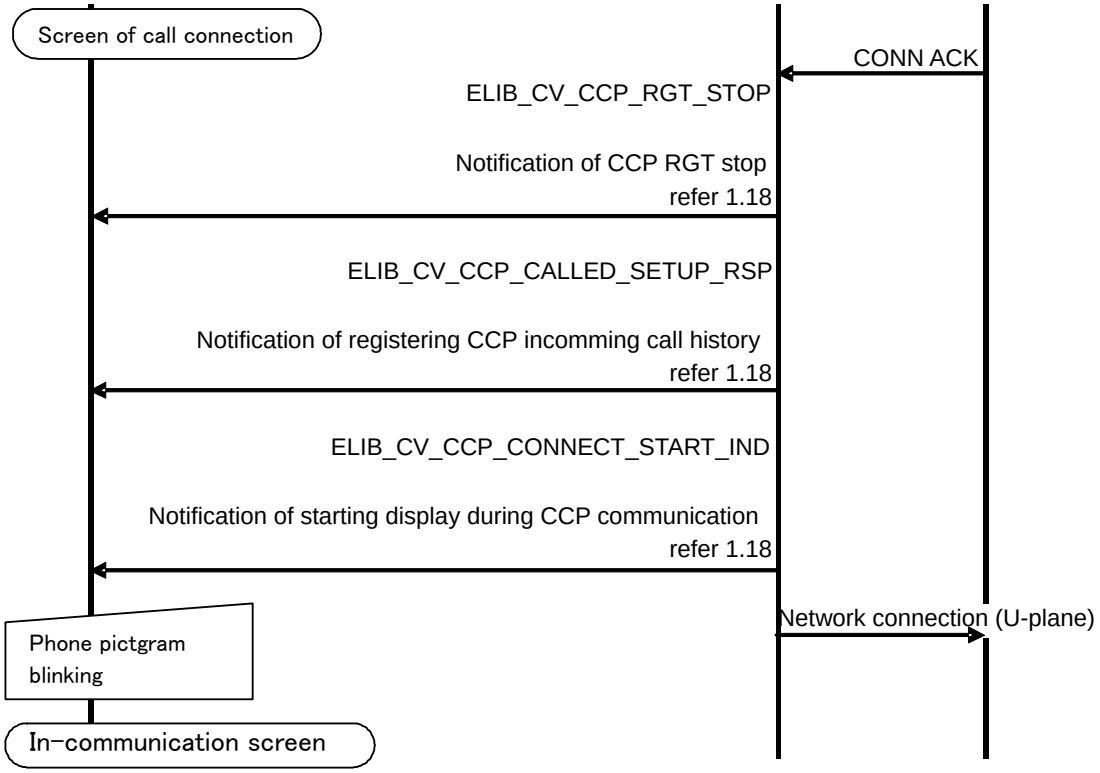


(2) Calling Sequence

a. Sequence from incoming status to conversation

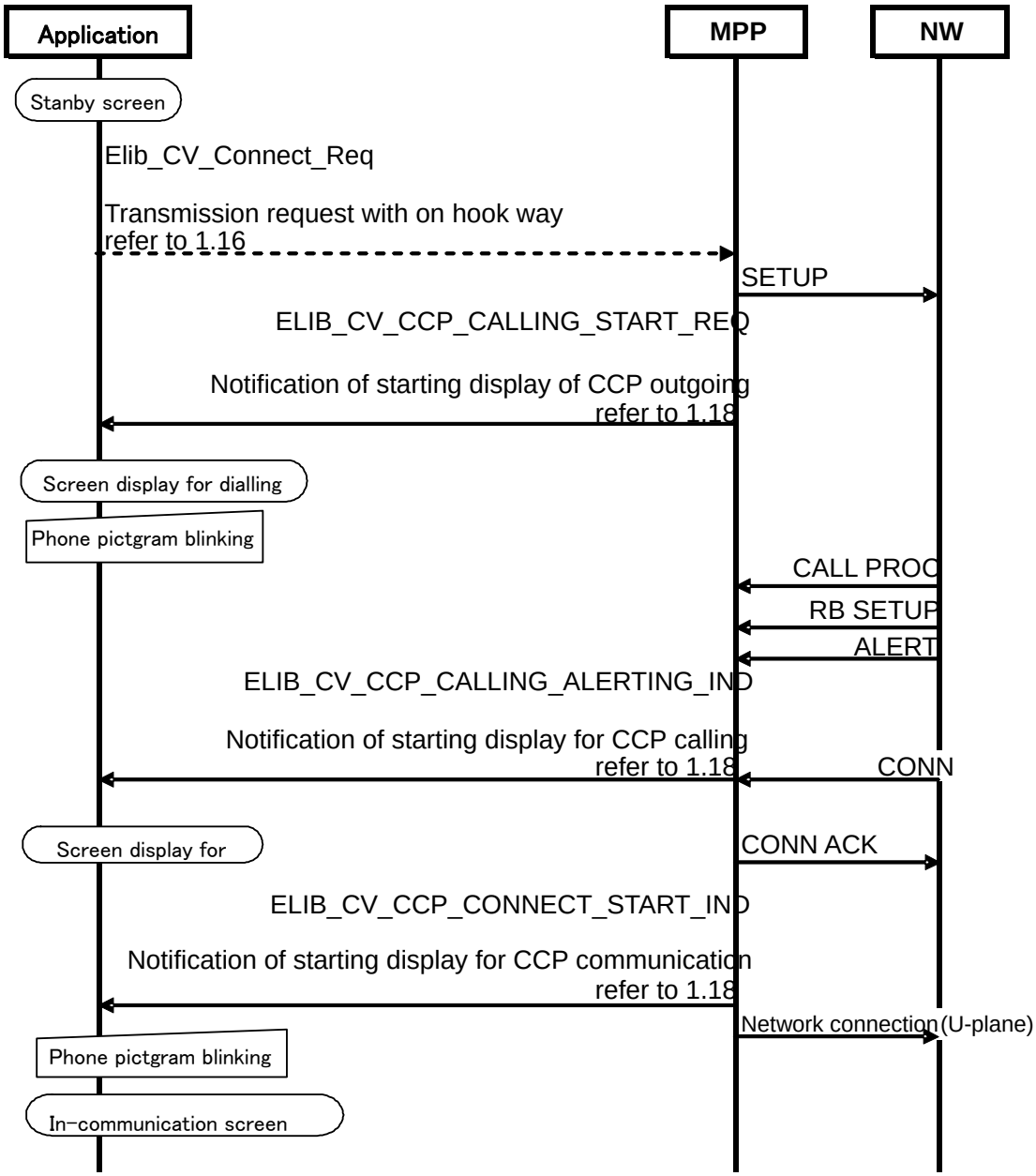
This is the sequence diagram that application transit to 'communicating' status by acknowledgement of incoming call notification..




Screen of call connection
CONN ACK
ELIB_CV_CCP_RGT_STOP
Notification of CCP RGT stop
refer 1.18
ELIB_CV_CCP_CALLED_SETUP_RSP
Notification of registering CCP incomming call history
refer 1.18
ELIB_CV_CCP_CONNECT_START_IND
Notification of starting display during CCP communication
refer 1.18
Network connection (U-plane)
Phone pictgram
blinking
In-communication screen

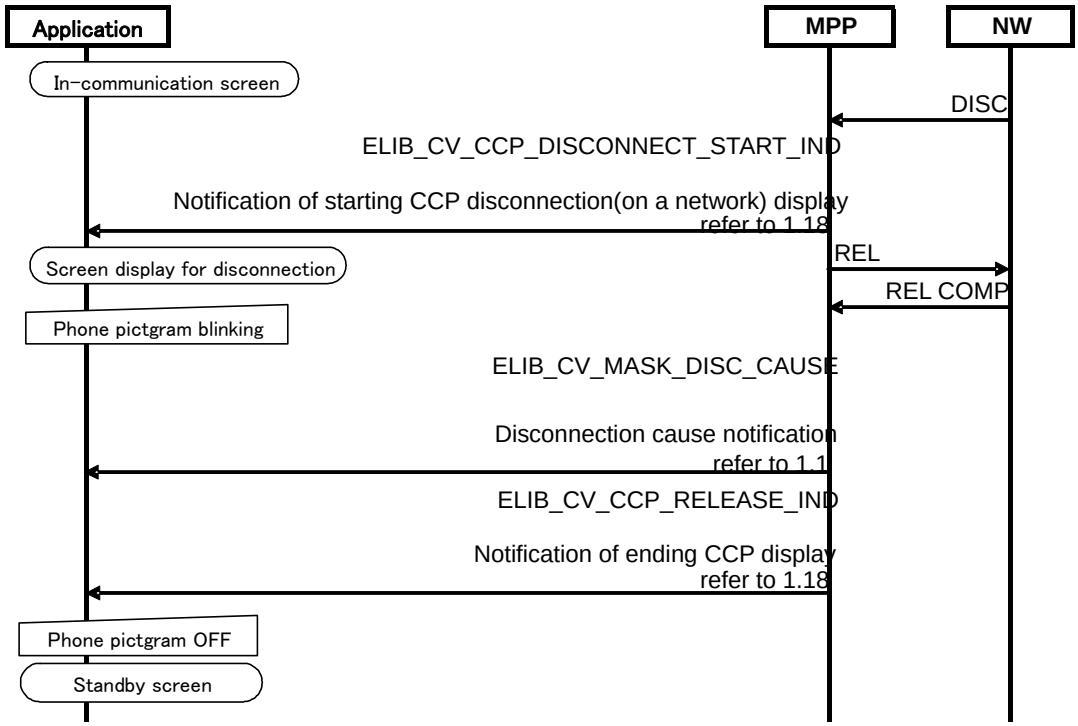
b. Sequence from outgoing status to conversation by on-hooking

This is the sequence diagram that application transit to a state of 'communicating' by acknowledgement of on-hook dialing notification.



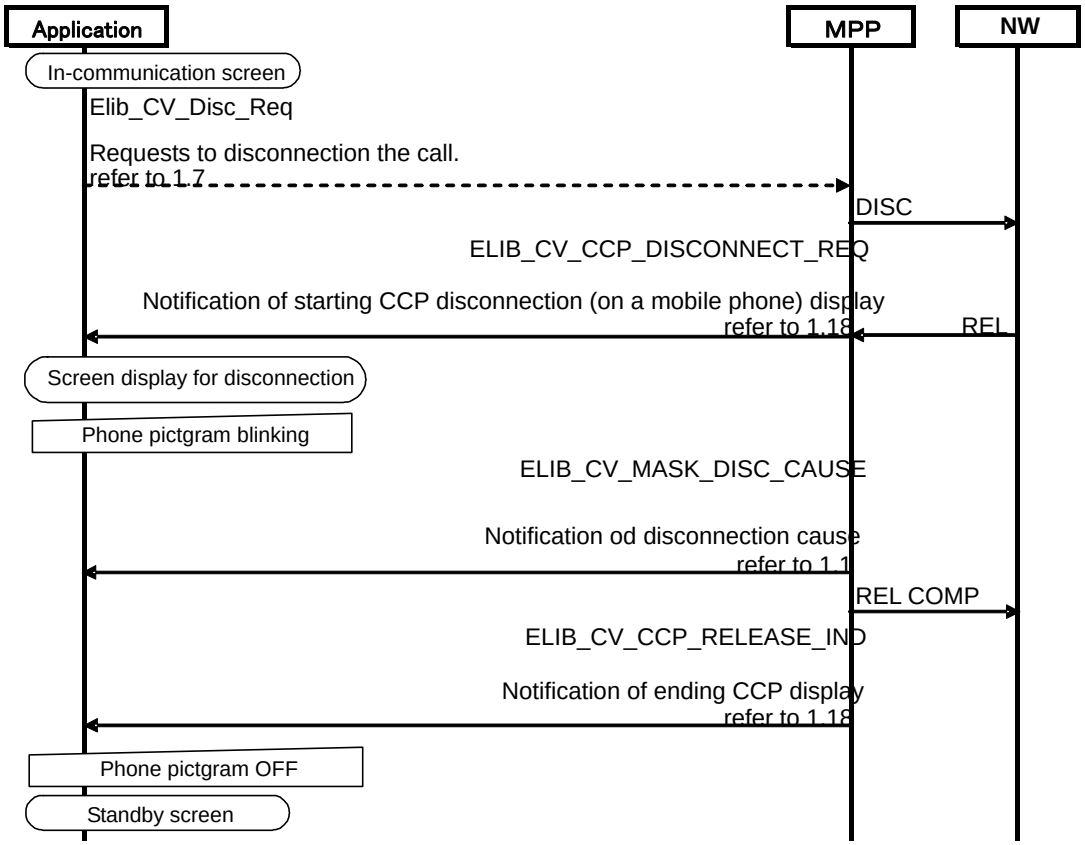
c. Sequence from conversation to disconnect by network

This is the sequence diagram that an application disconnects terminal by the network requested to disconnect while communication.



d. Sequence disconnecting communication by the request from mobile phone

This is the sequence diagram of the case that an application disconnects terminal by the request from terminal



3.1.1.2 Video Communication

Functional explanation

Video Communication is a one-to-one communication service of video using bi-directional or one-way 3G-324M, i.e. TV phone.

Function provided by MPP

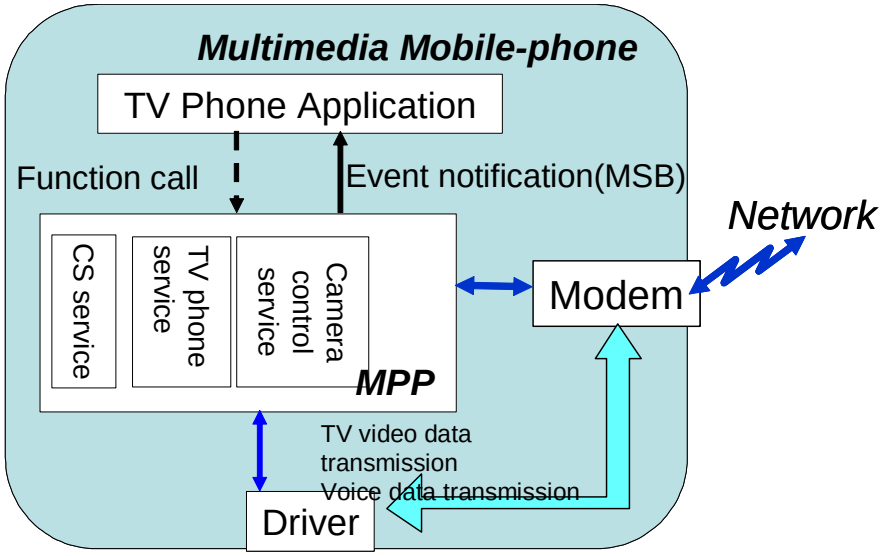
MPP provides following functionalities to develop applications that feature basic operations of TV phone (called TV phone AP).

- Communication using CS service which is network communication capability.
- TV phone service
- Camera control service
- Display communication status

Functional classification

Item #	functionality	Functional overview and examples
1	Dialing	• Perform dialing operation with video communication from mobile phone. • Perform generation of a dialing screen at communication status and response to user operation, etc.
2	Receiving incoming call	• Receive incoming call from video communication. • Perform generation of a dialing screen at communication status and response to user operation, etc..
3	Communication	• Sending/receiving voice data and TV picture data between driver and modem. • Perform displaying screen during video communication and mobile phone operation during displaying screen
4	Disconnection	• Perform disconnection operation of video communication from mobile phone. • Perform disconnection operation of video communication. • Perform displaying screen at disconnection and response to user operation, etc..

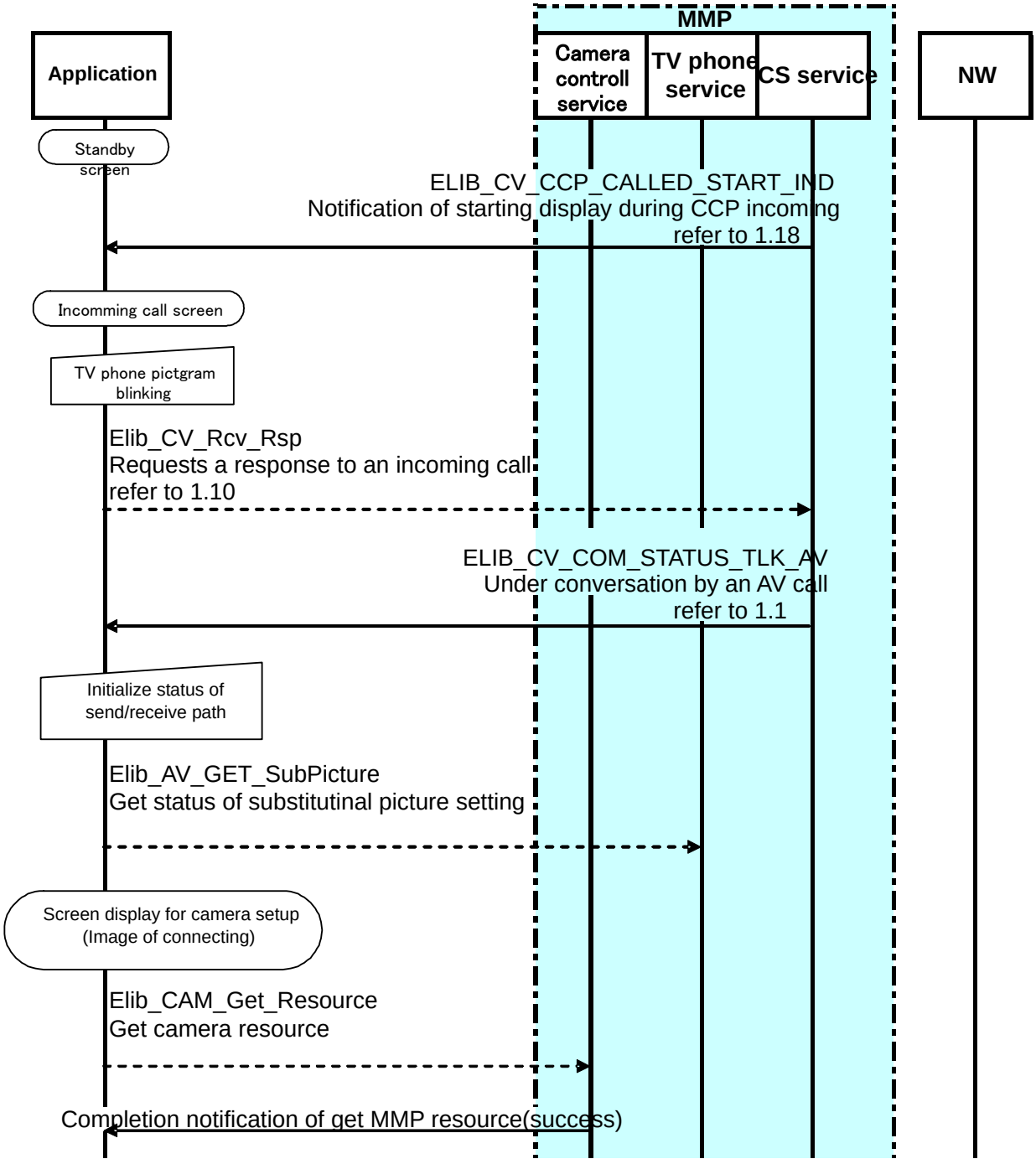
(1) Block Diagram

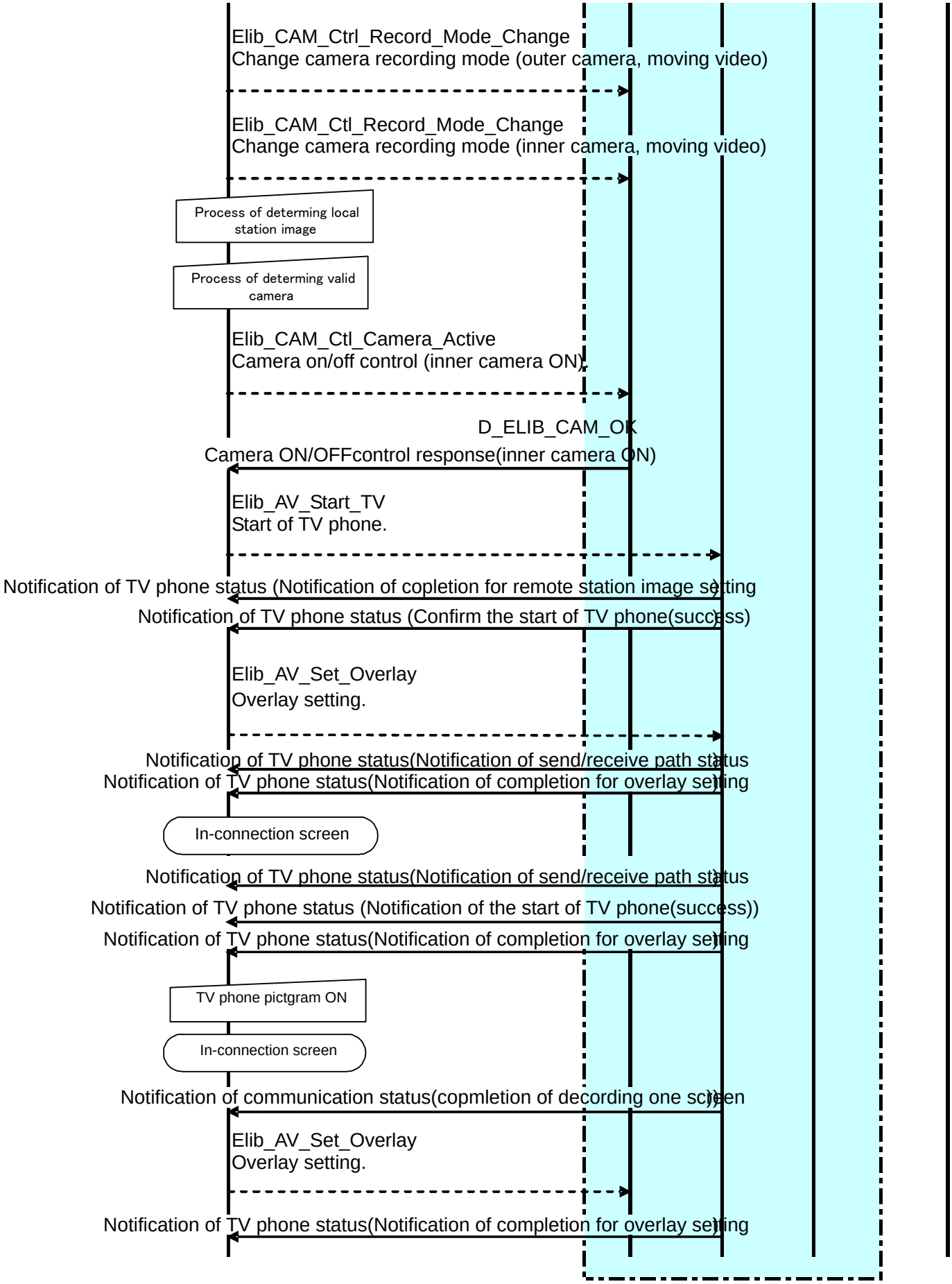


(2) Calling Sequence

a. Sequence from incoming status to conversation

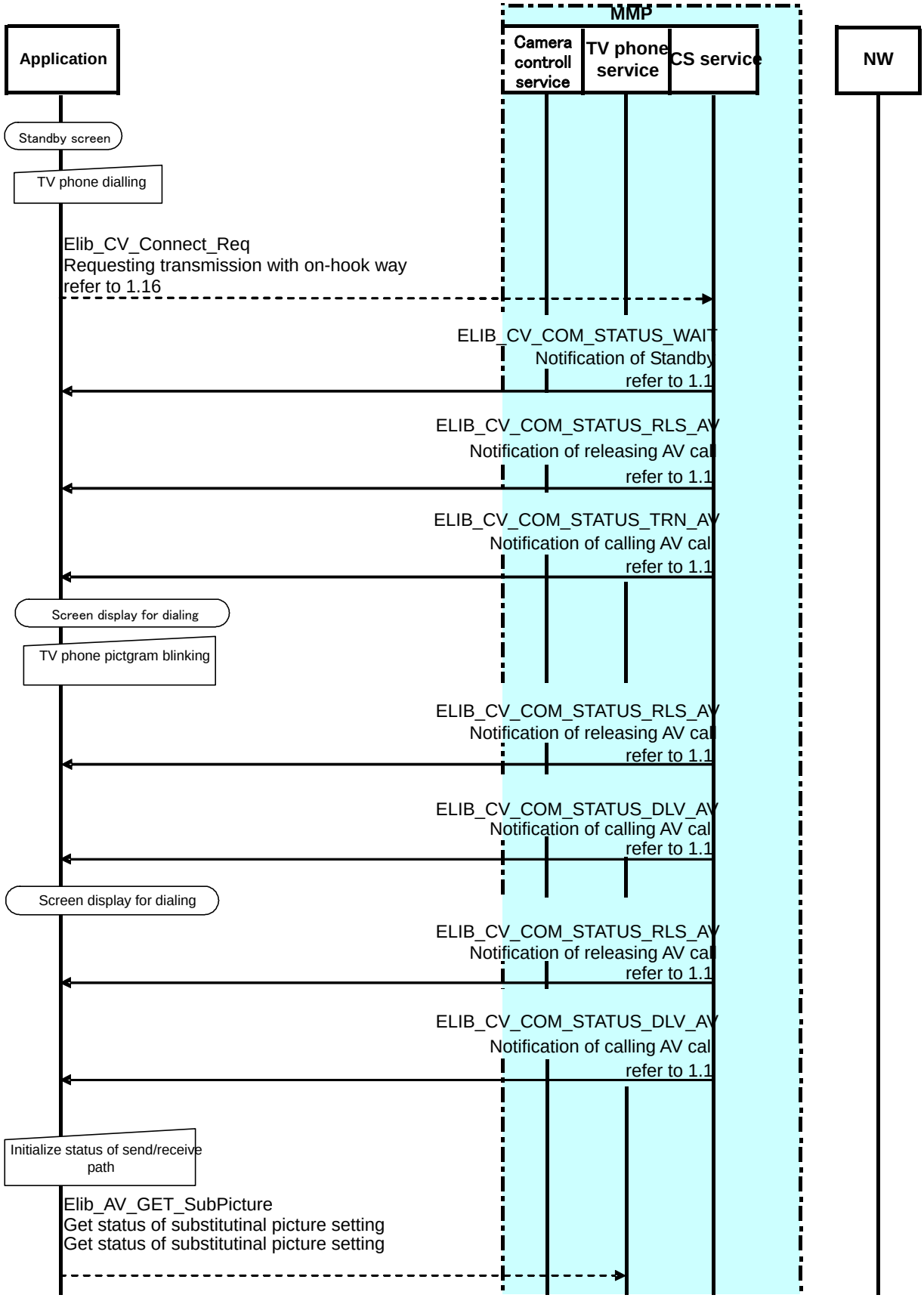
This is a sequence diagram that an application receives incoming call notification and transit to communicating status.

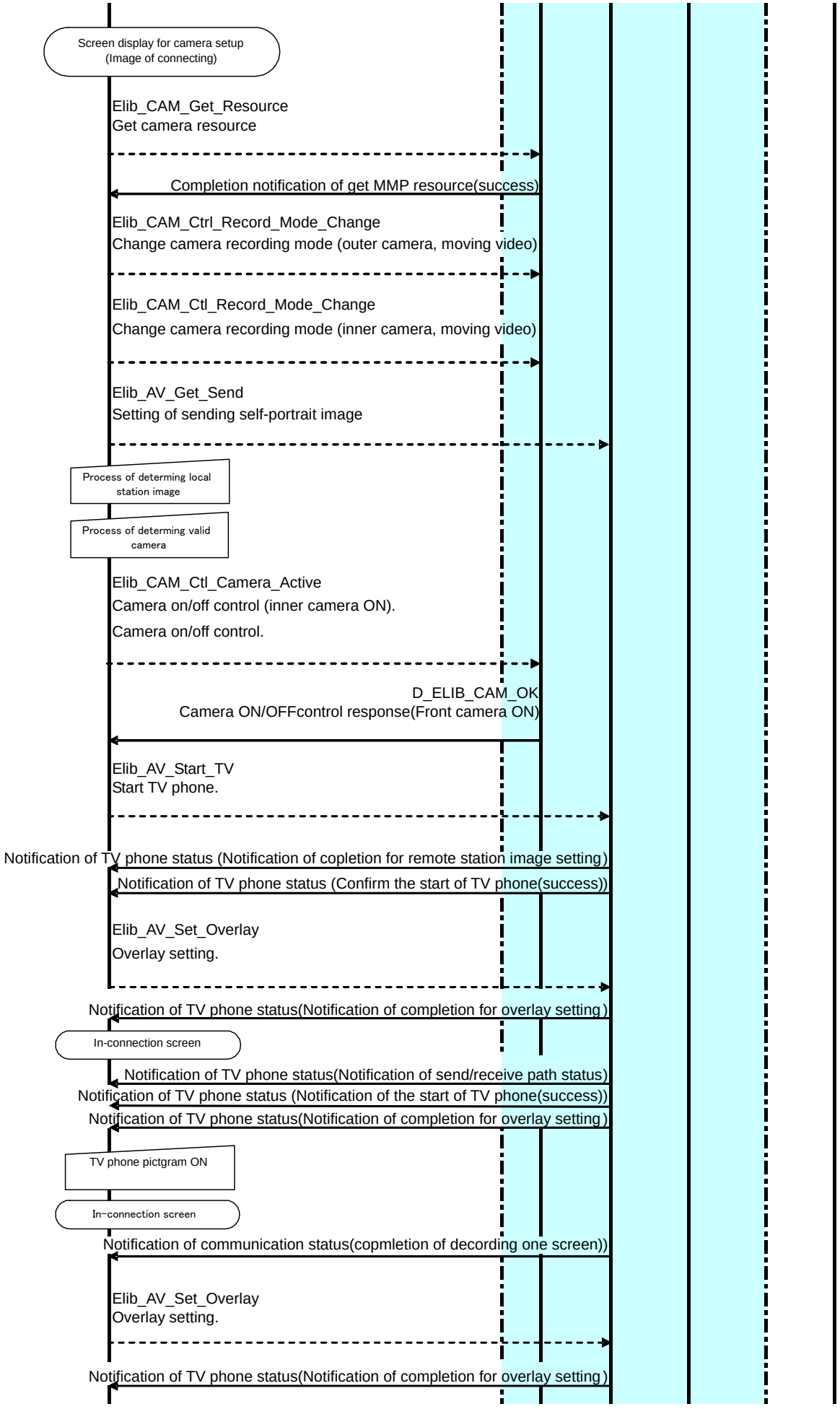



Elib_CAM_Ctrl_Record_Mode_Change
Change camera recording mode (outer camera, moving video)
Elib_CAM_Ctl_Record_Mode_Change
Change camera recording mode (inner camera, moving video)
Process of determing local station image
Process of determing valid camera
Elib_CAM_Ctl_Camera_Active
Camera on/off control (inner camera ON)
D_ELIB_CAM_OK
Camera ON/OFFcontrol response(inner camera ON)
Elib_AV_Start_TV
Start of TV phone.
Notification of TV phone status (Notification of copletion for remote station image setting
Notification of TV phone status (Confirm the start of TV phone(success)
Elib_AV_Set_Overlay
Overlay setting.
Notification of TV phone status(Notification of send/receive path status
Notification of TV phone status(Notification of completion for overlay setting
In-connection screen
Notification of TV phone status(Notification of send/receive path status
Notification of TV phone status (Notification of the start of TV phone(success))
Notification of TV phone status(Notification of completion for overlay setting
TV phone pictgram ON
In-connection screen
Notification of communication status(copmletion of decording one screen
Elib_AV_Set_Overlay
Overlay setting.
Notification of TV phone status(Notification of completion for overlay setting

b. Sequence from outgoing status to conversation

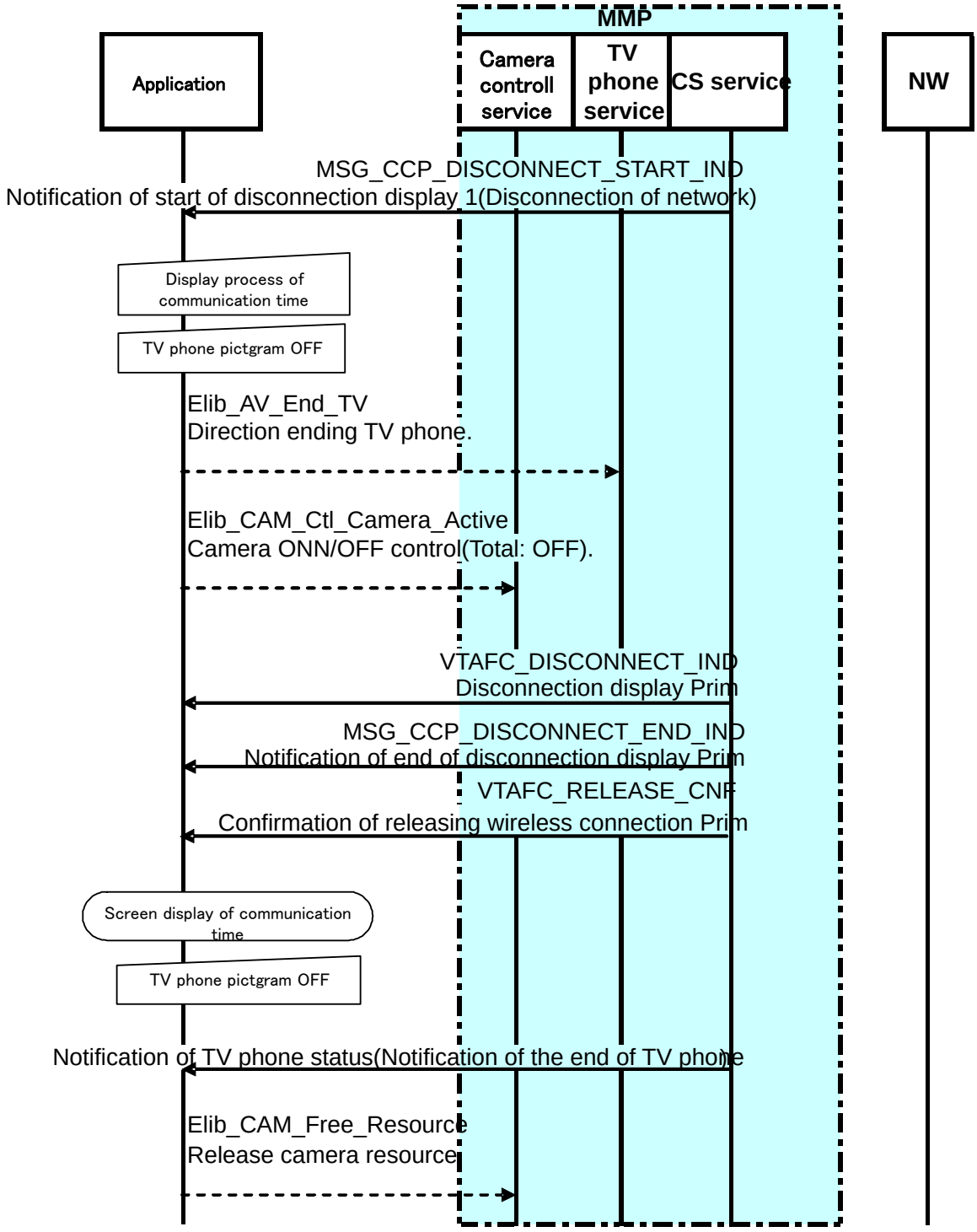
This is a sequence diagram that an application receives outgoing call notification and transit to communicating status.




Screen display for camera setup
(Image of connecting)
Elib_CAM_Get_Resource
Get camera resource
Completion notification of get MMP resource(success)
Elib_CAM_Ctrl_Record_Mode_Change
Change camera recording mode (outer camera, moving video)
Elib_CAM_Ctl_Record_Mode_Change
Change camera recording mode (inner camera, moving video)
Elib_AV_Get_Send
Setting of sending self-portrait image
Process of determing local station image
Process of determing valid camera
Elib_CAM_Ctl_Camera_Active
Camera on/off control (inner camera ON).
Camera on/off control.
D_ELIB_CAM_OK
Camera ON/OFFcontrol response(Front camera ON)
Elib_AV_Start_TV
Start TV phone.
Notification of TV phone status (Notification of copletion for remote station image setting)
Notification of TV phone status (Confirm the start of TV phone(success))
Elib_AV_Set_Overlay
Overlay setting.
Notification of TV phone status(Notification of completion for overlay setting)
In-connection screen
Notification of TV phone status(Notification of send/receive path status)
Notification of TV phone status (Notification of the start of TV phone(success))
Notification of TV phone status(Notification of completion for overlay setting)
TV phone pictgram ON
In-connection screen
Notification of communication status(copmletion of decording one screen))
Elib_AV_Set_Overlay
Overlay setting.
Notification of TV phone status(Notification of completion for overlay setting)

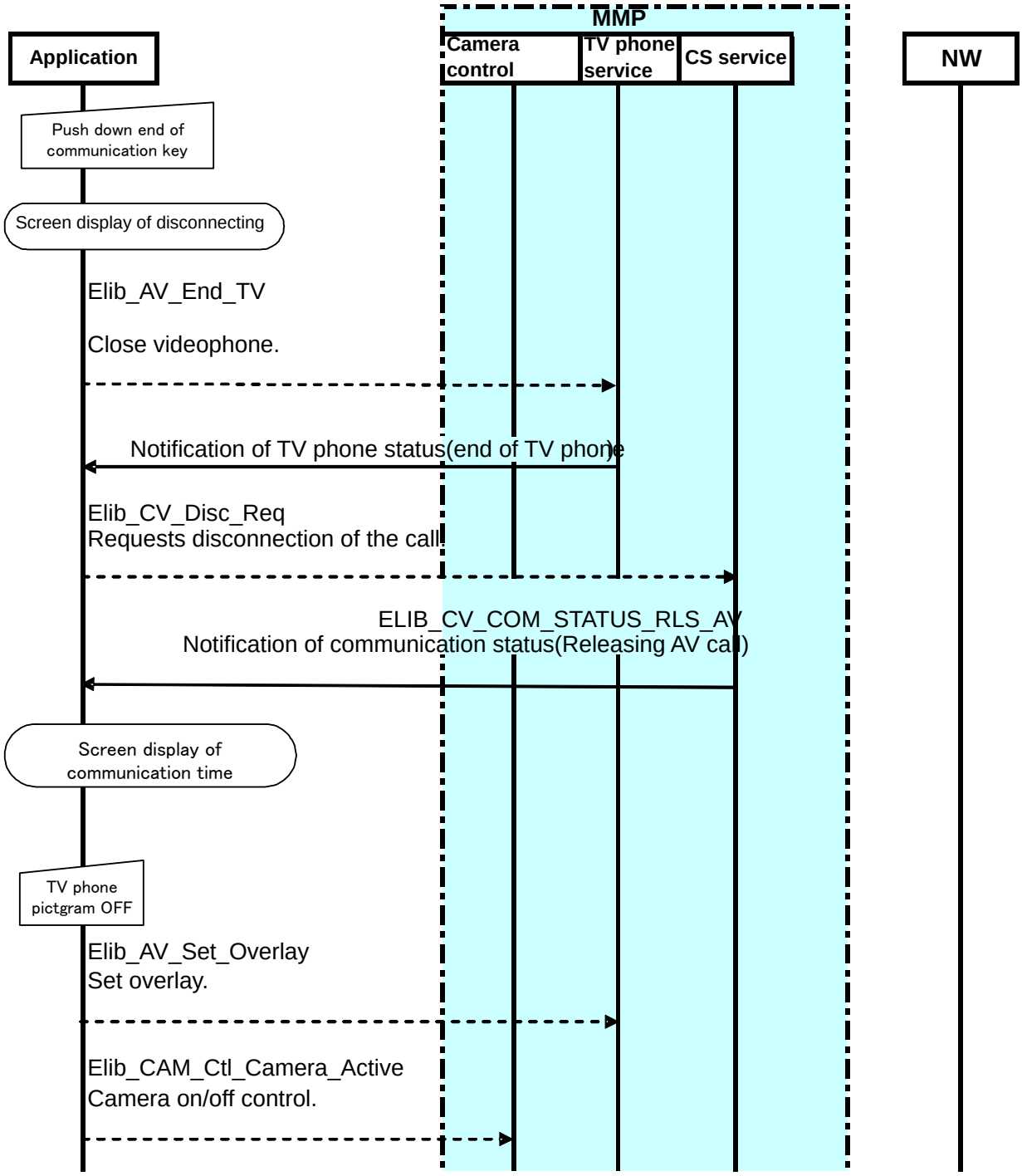
c. Sequence from conversation to disconnect by network

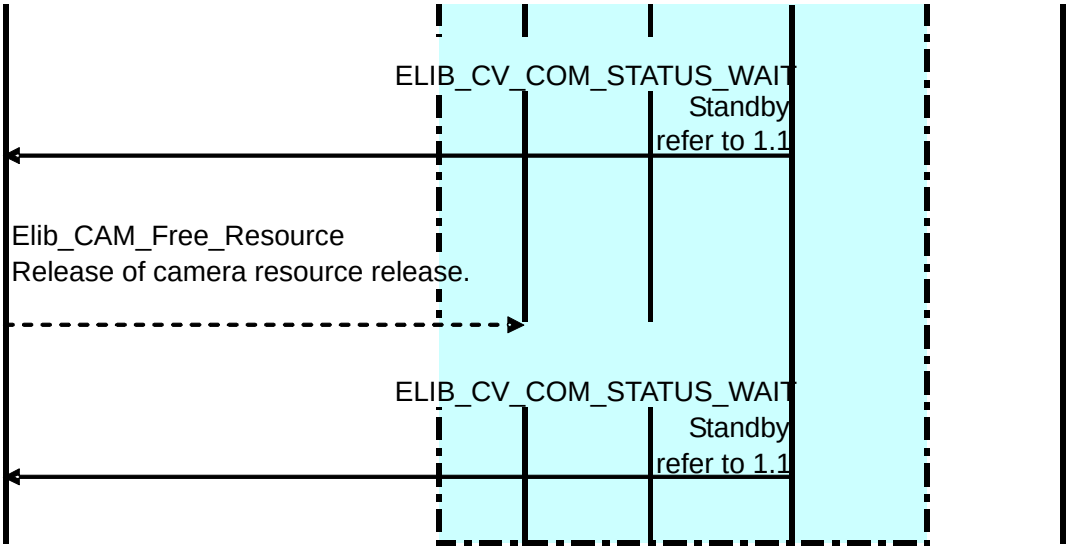
This is a sequence diagram that an application receives disconnect request from network and disconnect a terminal



d. Sequence from conversation to disconnect by mobile phone

This is a sequence diagram that, in communication status, an application receives disconnect request from TV phone and transit to disconnects TV phone.




ELIB_CV_COM_STATUS_WAIT
Standby
refer to 1.1
Elib_CAM_Free_Resource
Release of camera resource release.
ELIB_CV_COM_STATUS_WAIT
Standby
refer to 1.1

3.1.1.3 Unrestricted Digital Data Communication

Functional explanation

Unrestricted Digital Data Communication is a high quality and bit transparent end-to-end data communication service using line switching.
External TE (i.e. External device such as PC) connected to mobile phone using USB can send/receive data to/from the network by controlling mobile phones using AT command with unrestricted digital communication.

Function provided by MPP

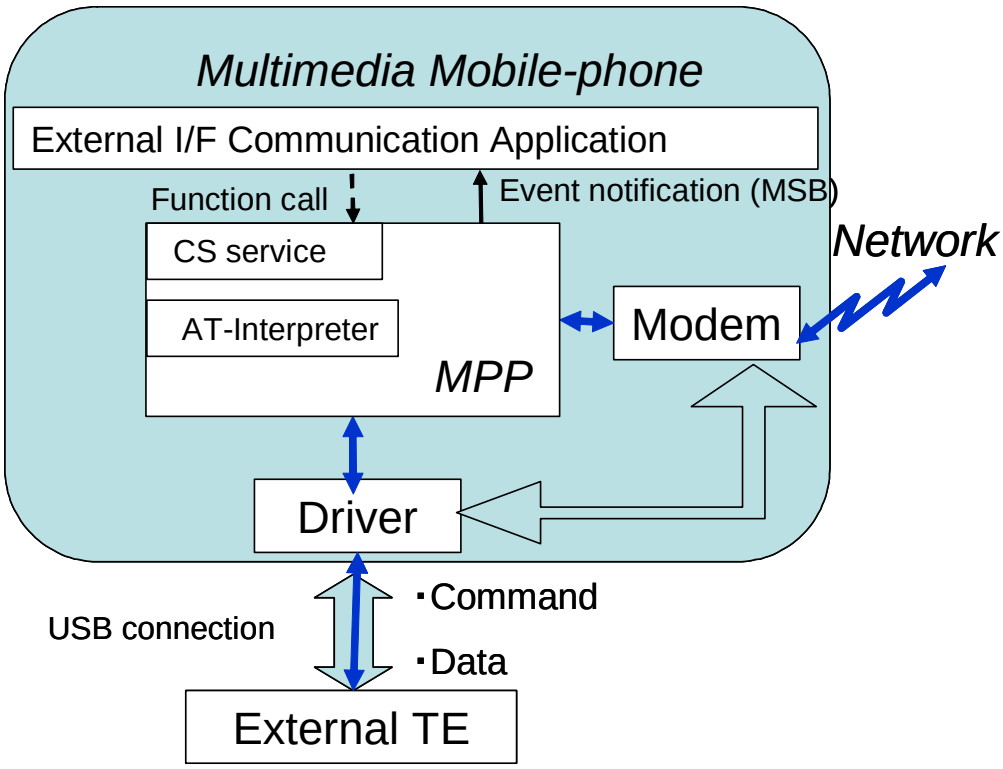
MPP provides following functionalities.
- Perform unrestricted digital communication
- Communicate with external TE using AT command.
- Display the status of communication related to both communications.

Details of communication with external TE should be referred (2)-b.

Functional classification

Item #	functionality	Functional overview and examples
1	Dialing	・Perform dialing operation with unrestricted digital communication from mobile phone or external device. ・During above dialing operation, display icon on mobile screen indicating the destination phone number and that dialing is taking place.
2	Receiving incoming call	・Receive incoming call from unrestricted digital communication. ・During above receiving incoming call, display icon on mobile screen indicating the phone number of sender and that receiving incoming call is taking place.
3	Communication	・Sending/receiving data between driver and modem. ・On mobile screen, display string and icon indicating that communication is taking place, and the communication duration.
4	Disconnection	・Perform disconnection operation of unrestricted digital communication from mobile phone or external device. ・Perform disconnection operation of unrestricted digital communication from network. ・During above disconnection operation, display icon on mobile screen indicating that disconnection is taking place.

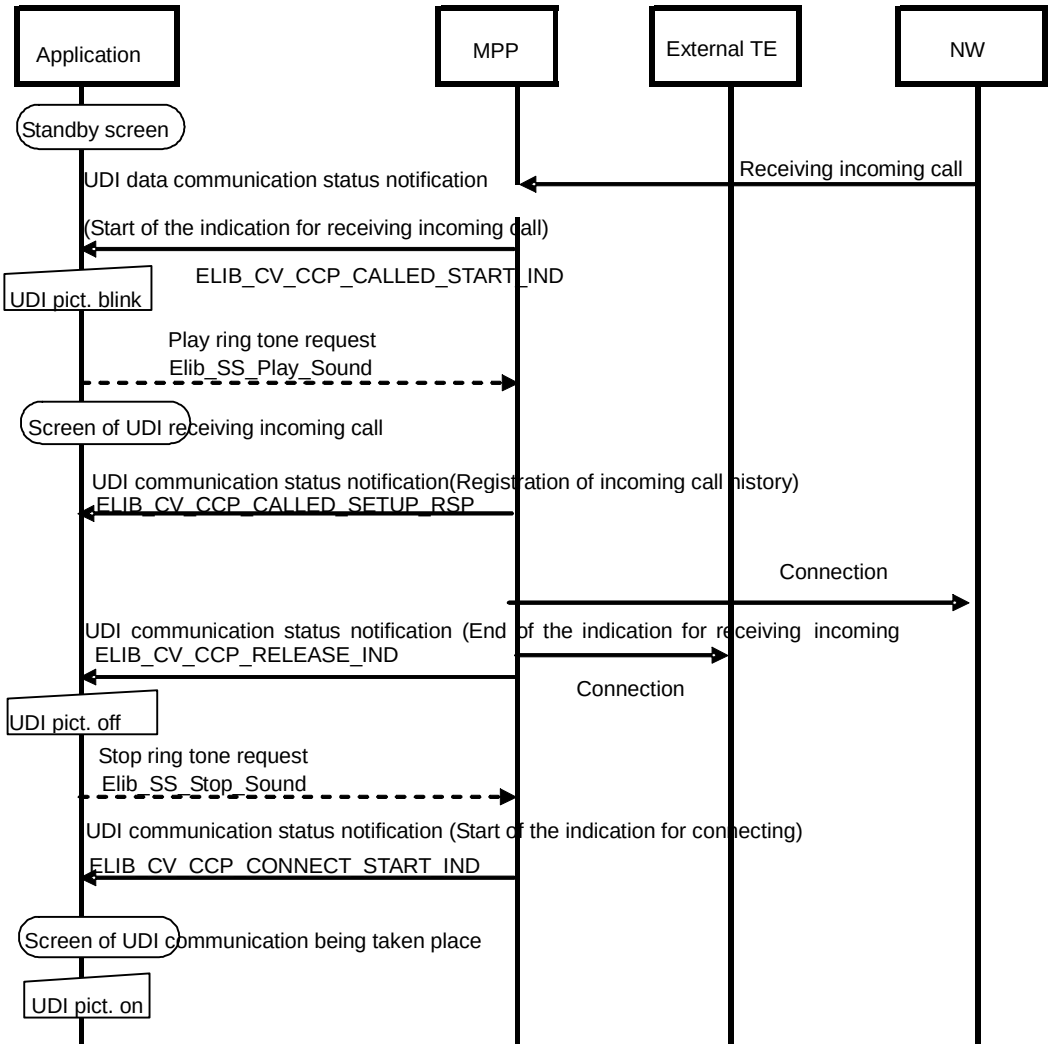
(1) Block Diagram



(2) Calling Sequence

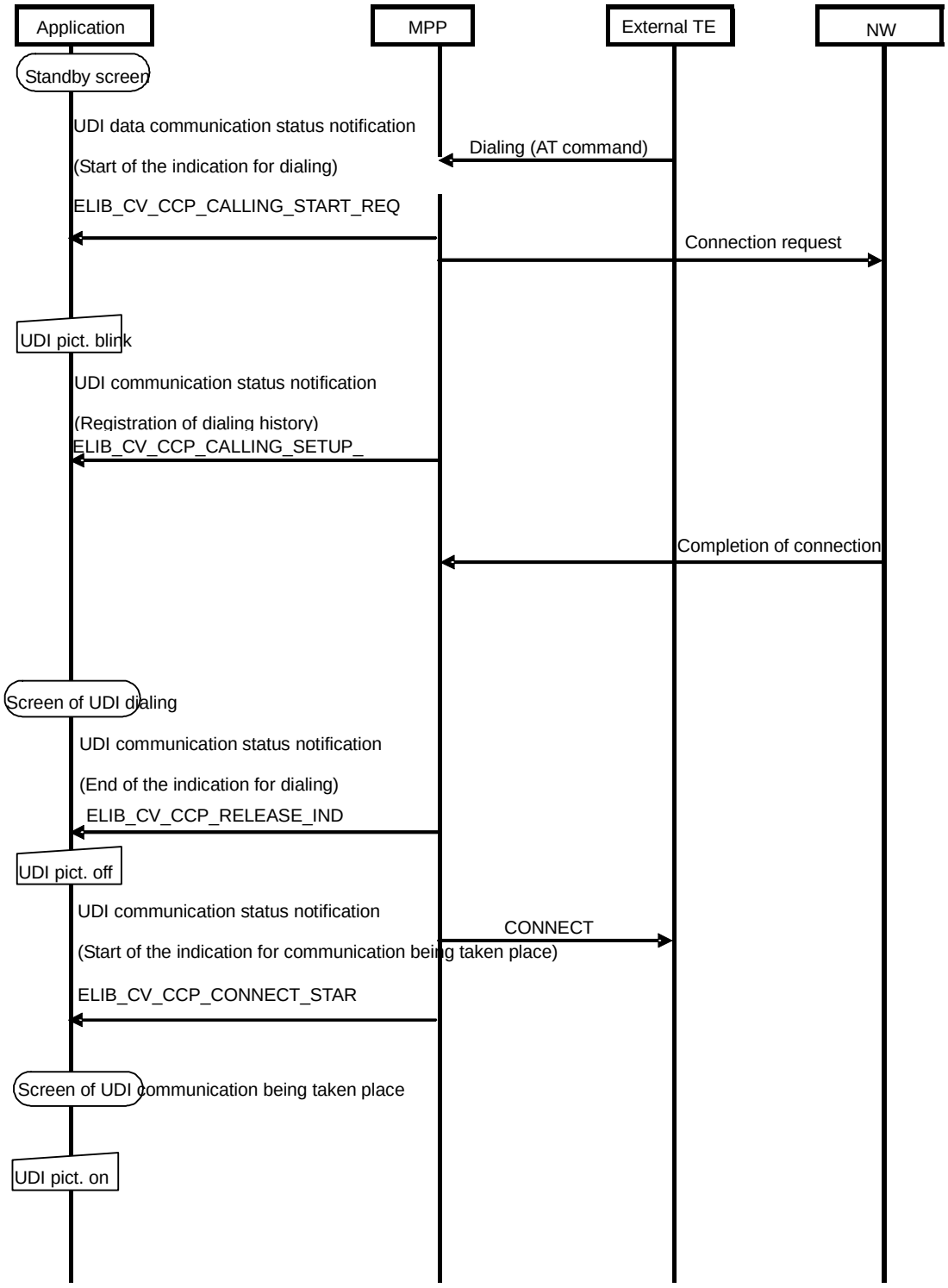
a. Sequence from incoming status to conversation

This is a sequence diagram indicating the status transition from receiving incoming call notification with UDI to communication status.



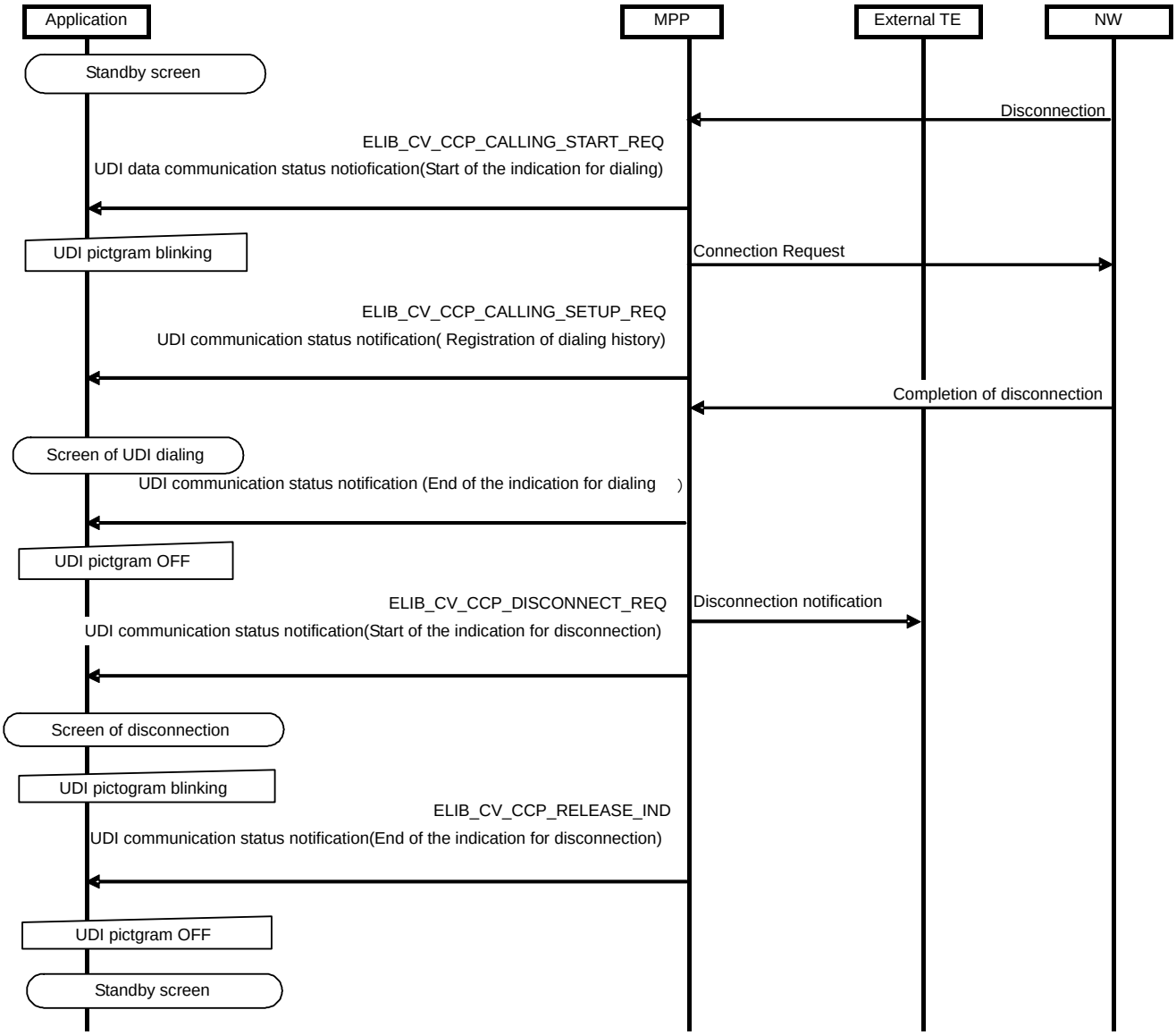
b. Sequence from originating status to conversation

This is a sequence diagram indicating the status transition from receiving the notification of UDI dialing by external TE to communication status.



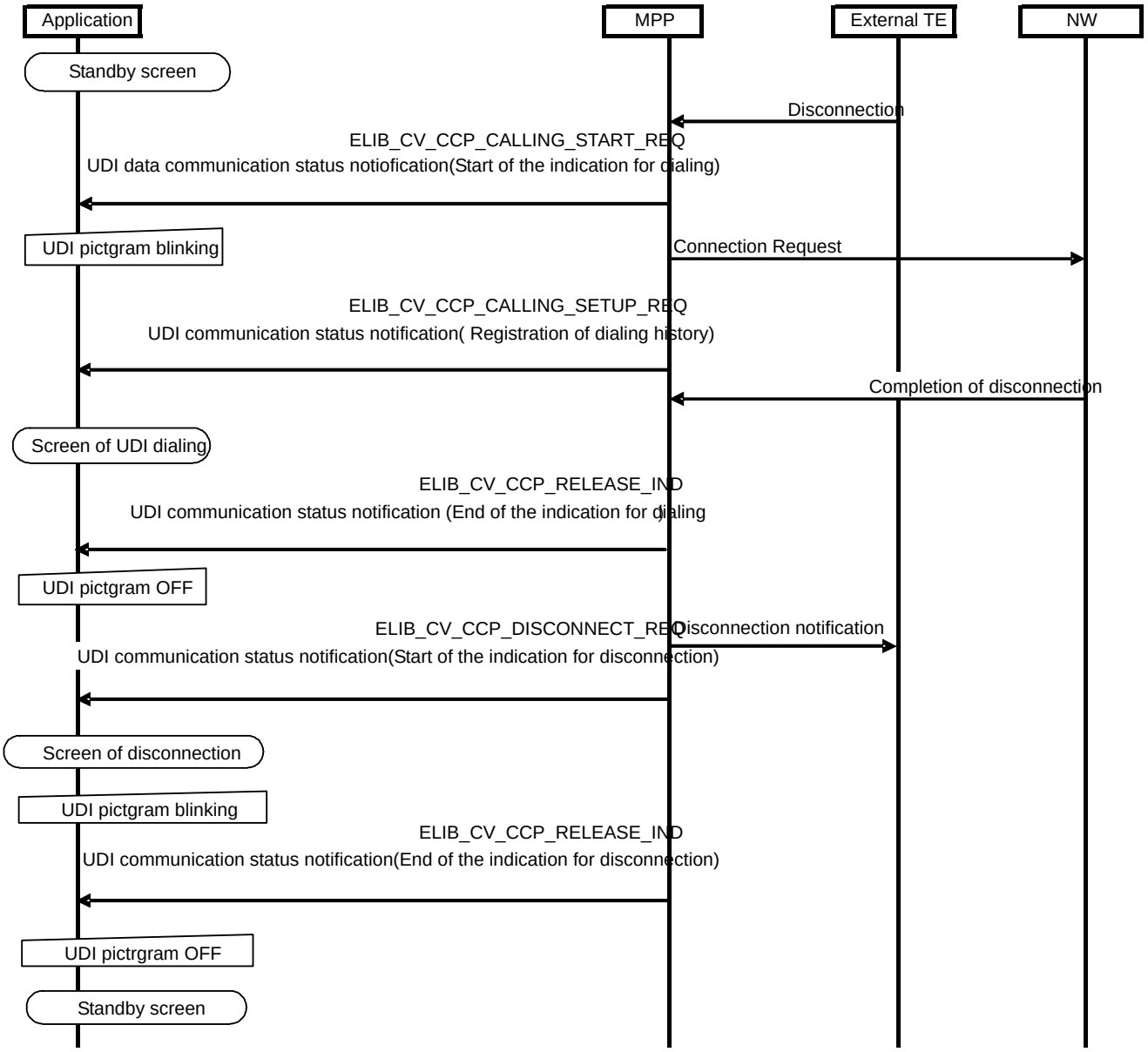
c. Sequence from conversation to disconnect by network

This is a sequence diagram indicating the status transition, by receiving the disconnection request from network , from communication status to disconnection status.



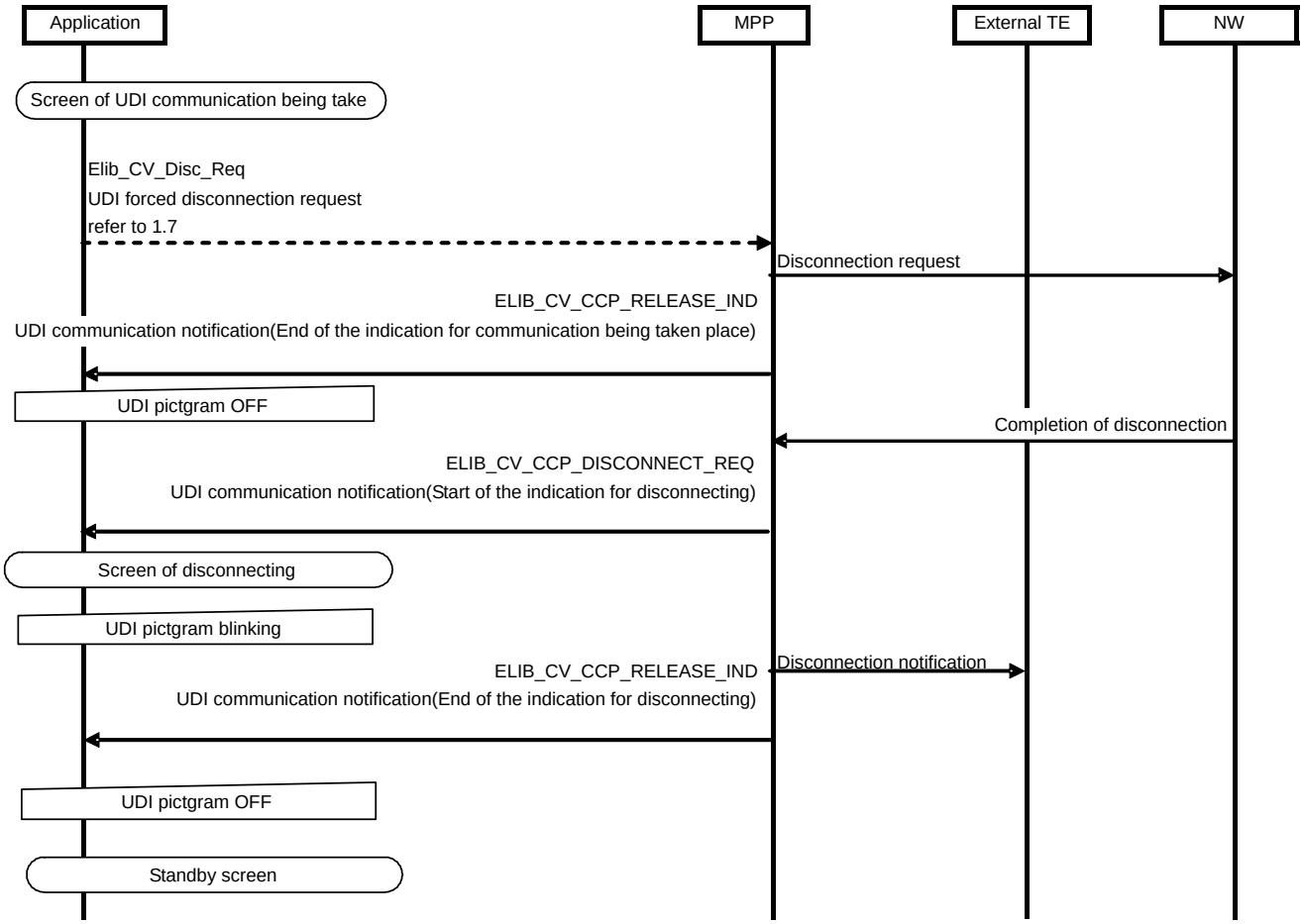
d. Sequence from conversation to disconnect by external TE

This is a sequence diagram indicating external TE requests disconnection in communication status.



e. Sequence from conversation to disconnect by mobile phone

This is a sequence diagram indicating terminal requests disconnection in communication status.

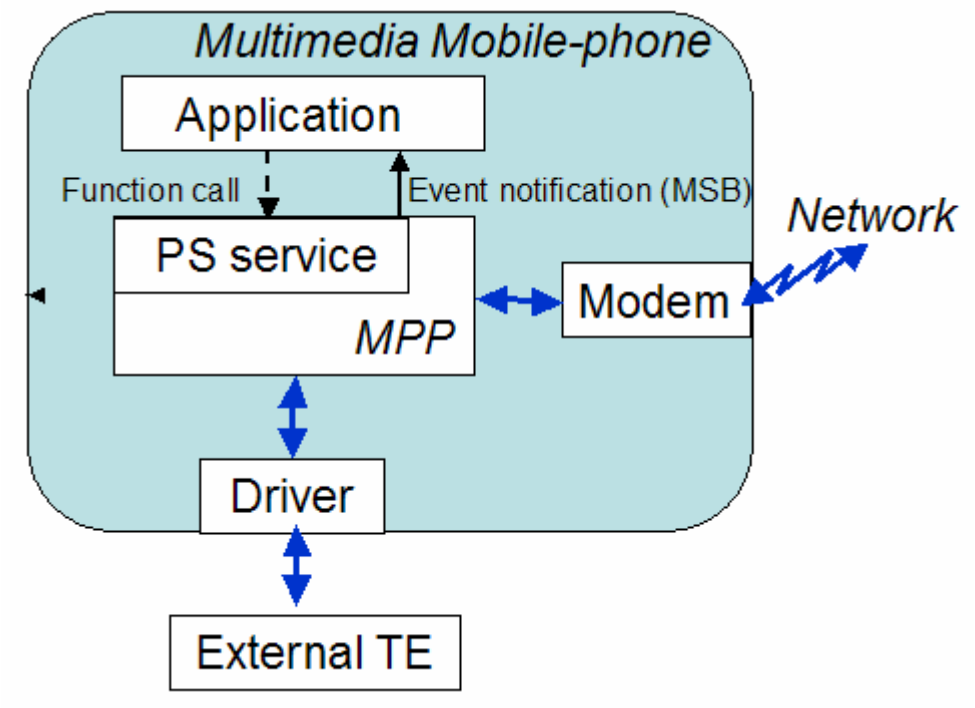


3.1.2 Packet Switched Communication Service

3.1.2.1 PPP dial-up communication service

PPP dial-up communication service is a packet communication service with PPP. Typically it is a functionality that enables PPP packet communication with external TE using external device, such as USB.
MPP provides functionalities to enable PPP dial-up communication services.

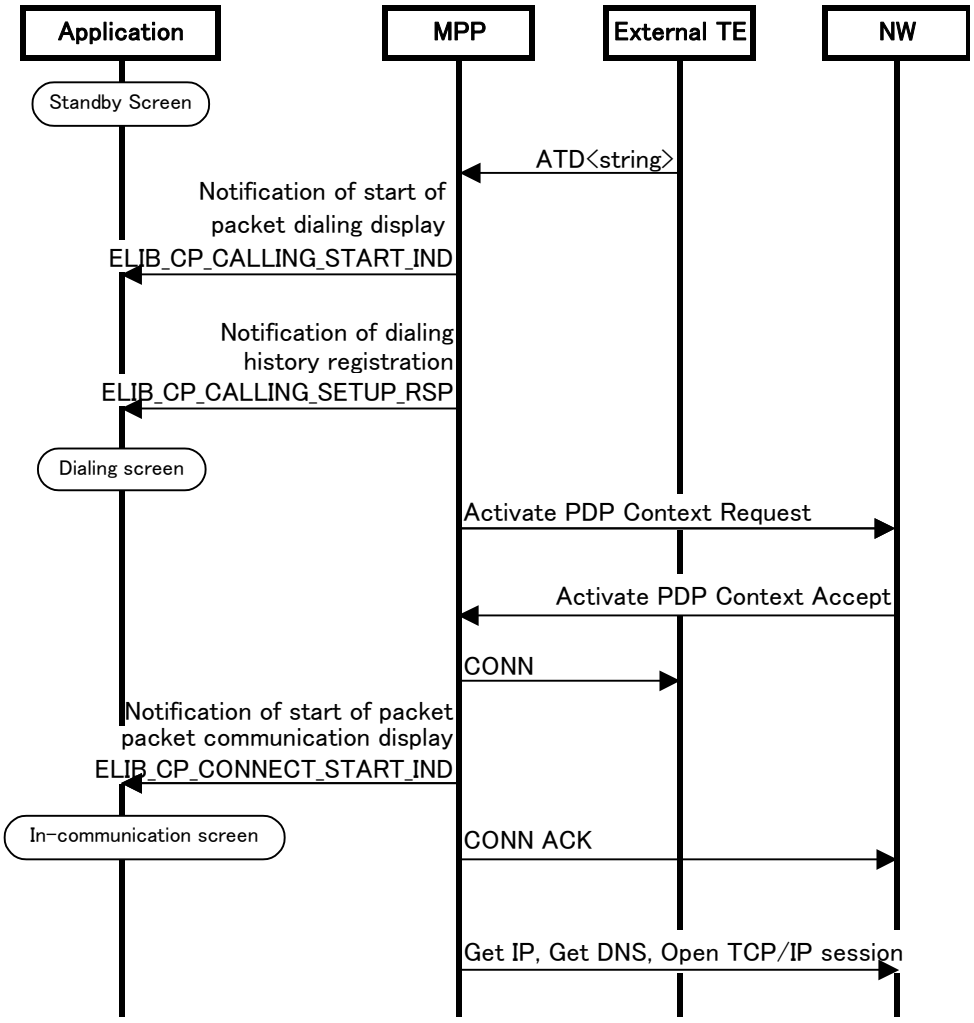
(1)Block Diagram



(2)Sequence

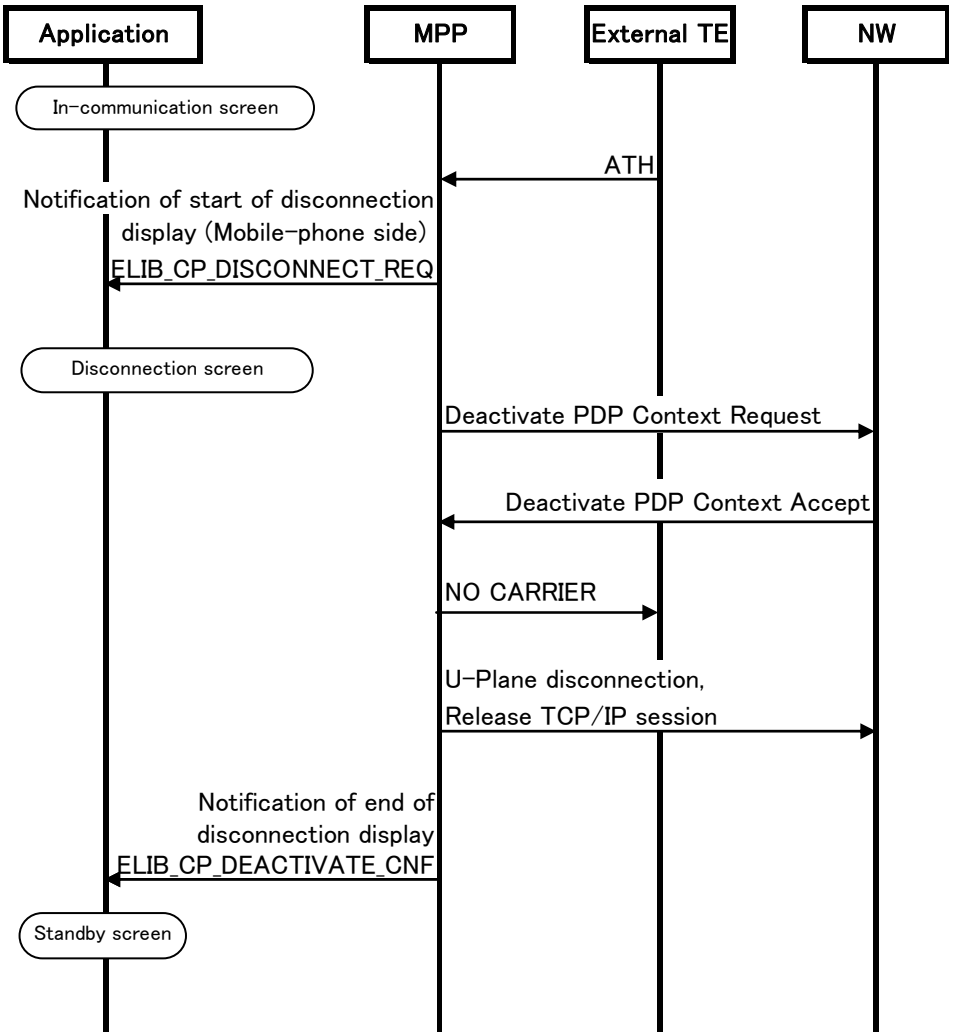
a. Dialing action from external device

Following diagram shows a sample sequence from receiving dialing request from external device to establishing PPP link.



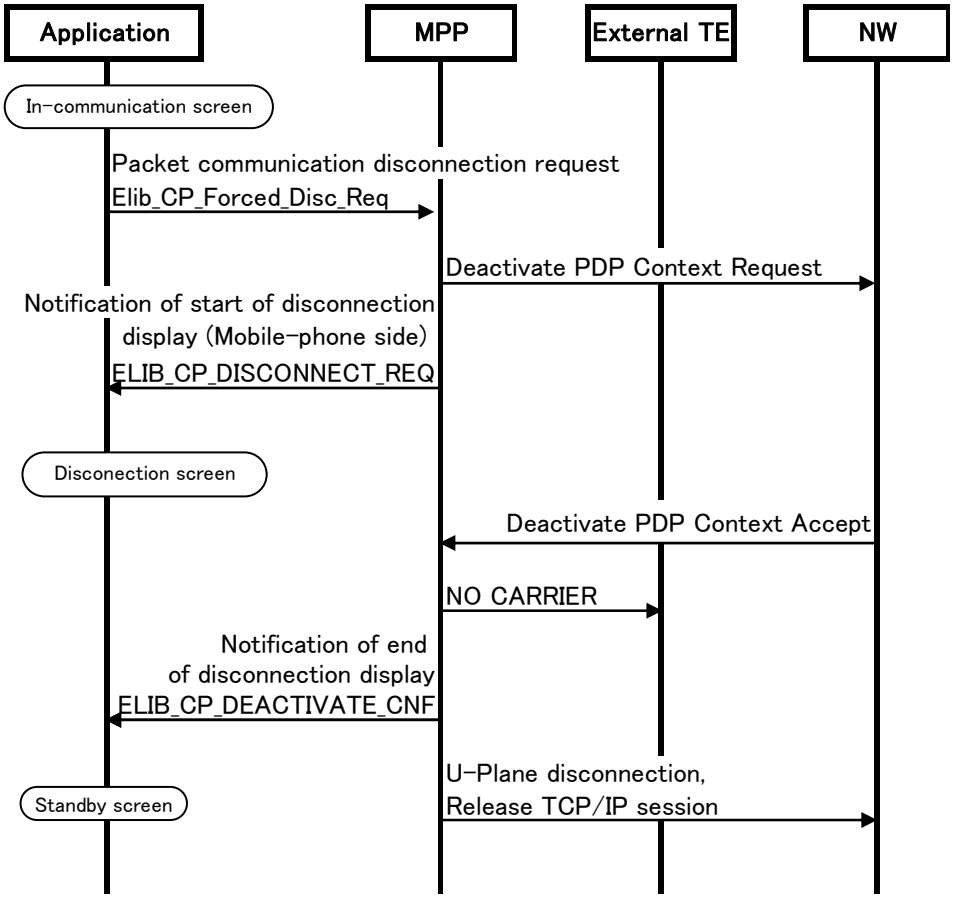
b. Disconnecting action from external device

Following diagram shows a sample sequence to take disconnection action by receiving disconnection request from external device.



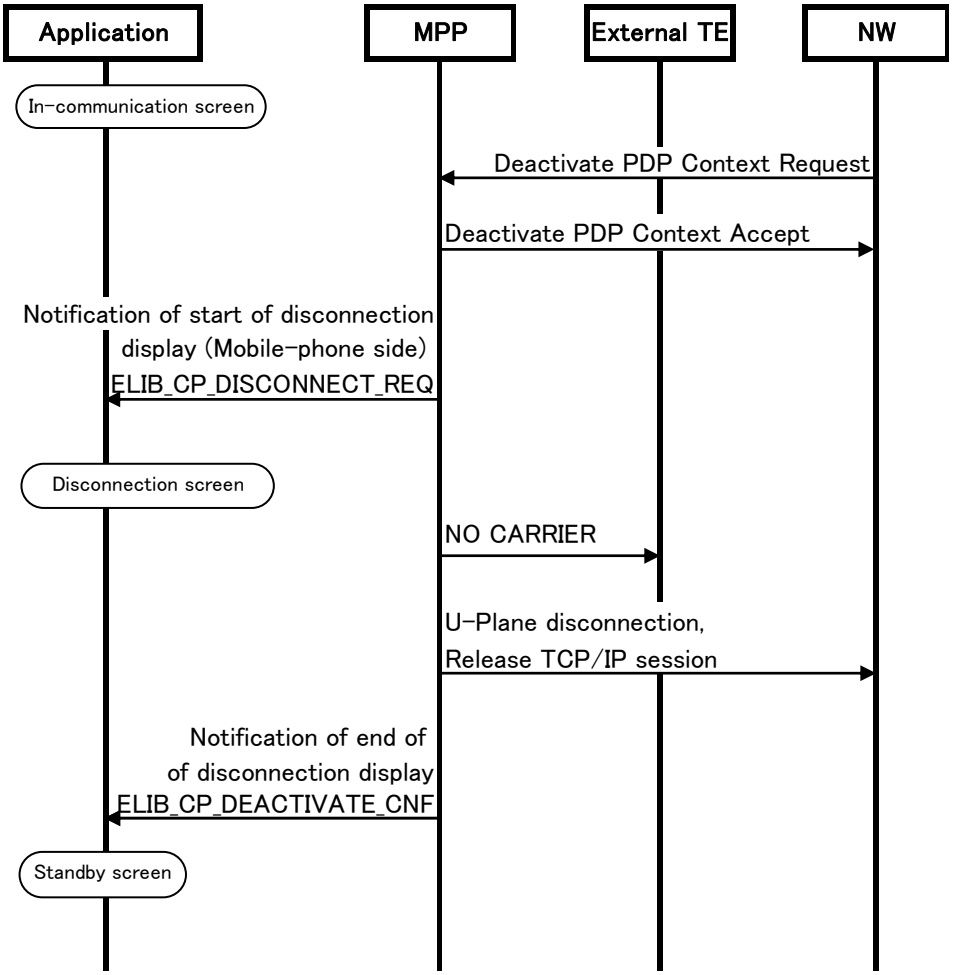
c. Disconnect from mobile-phone in communication status

Following diagram shows a sample sequence to disconnect from mobile-phone in communication status.



d. Disconnect from NW side in communication status

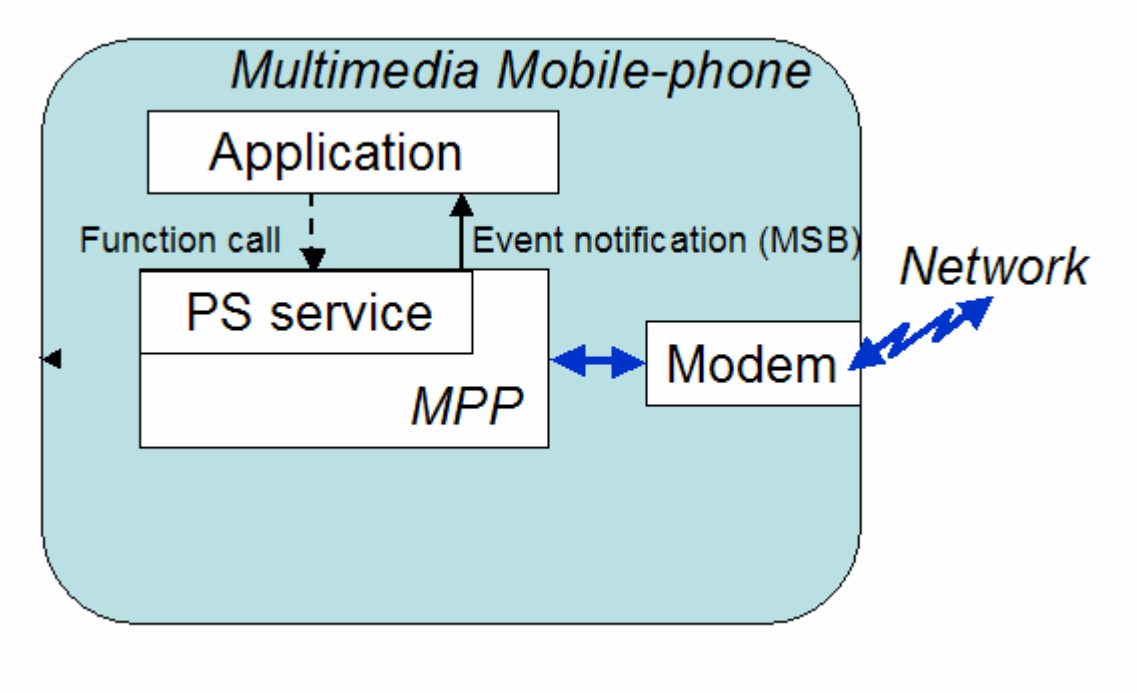
Following diagram shows a sample sequence to disconnect from NW side in communication status.



3.1.2.2 IP Connection type of data transfer service

IPconnection type of data transfer service is a service not using PPP.
MPP provides functionalities to enable IP connection type of data transfer service.

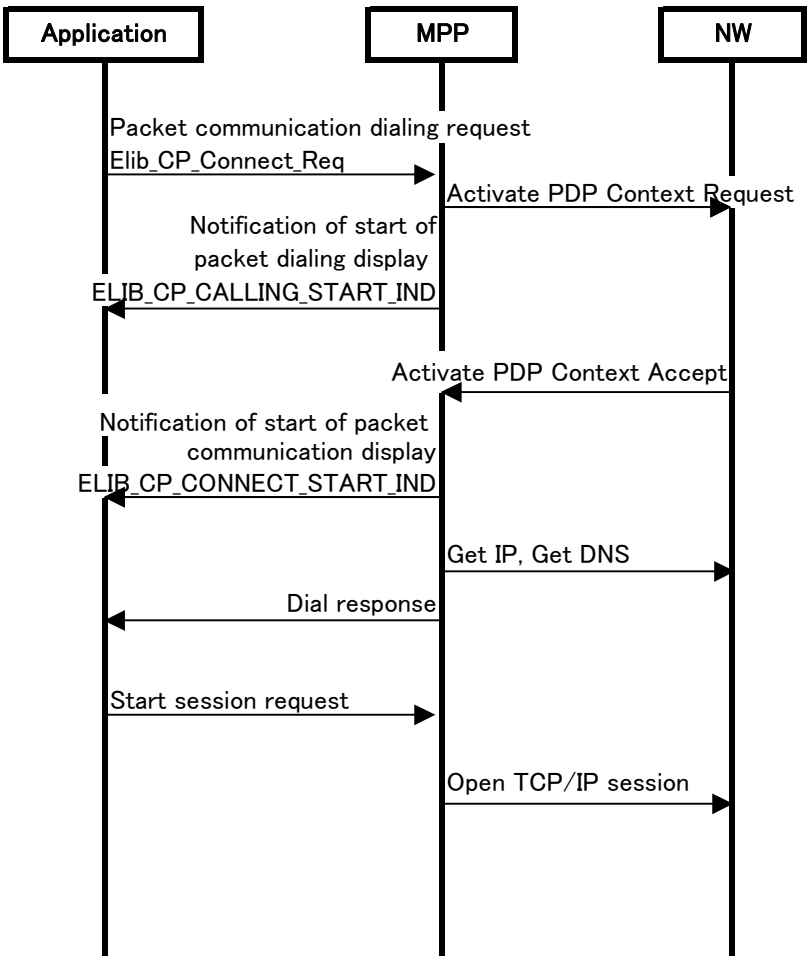
(1)Block diagram



(2)Sequence

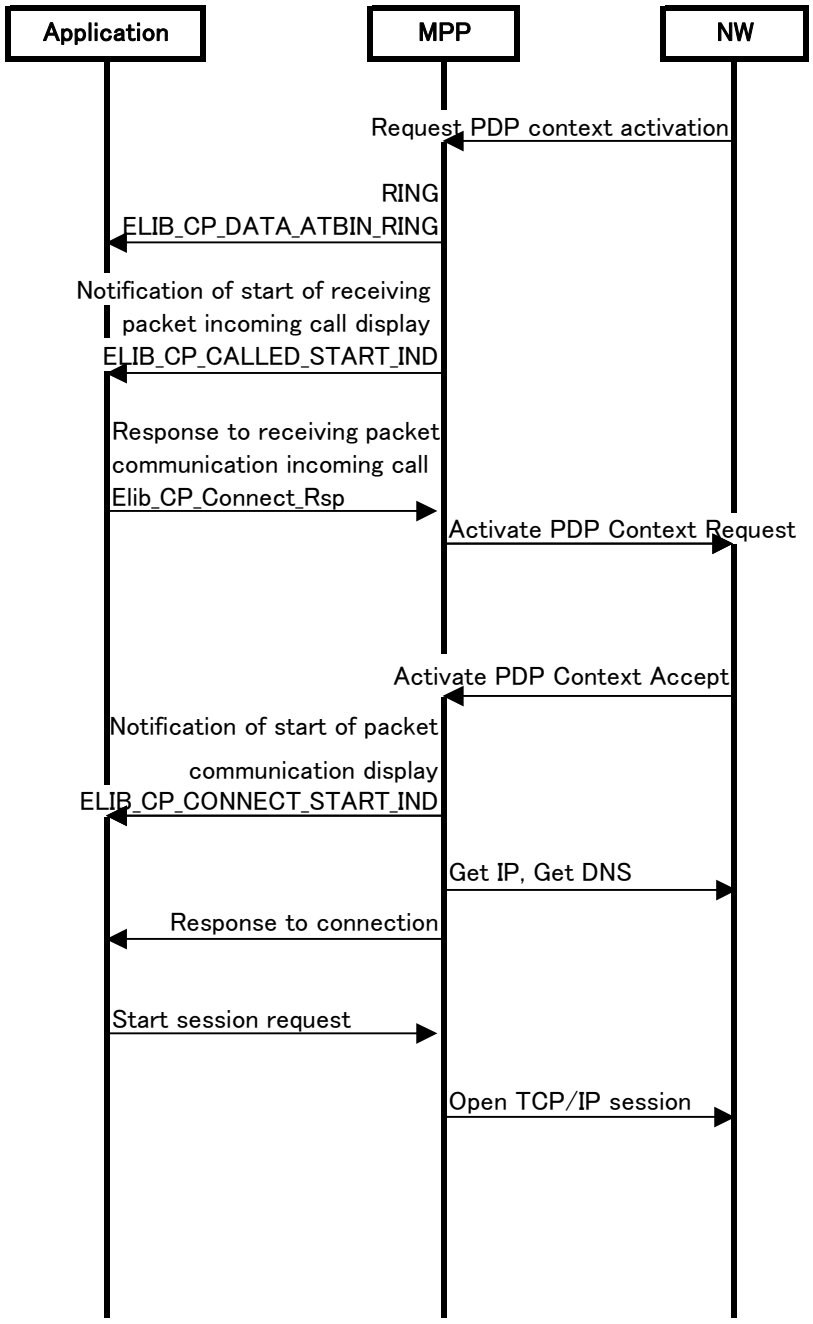
a. Dialing action from mobile-phone

Following diagram shows a sample sequence from issuing dialing request to establish service.



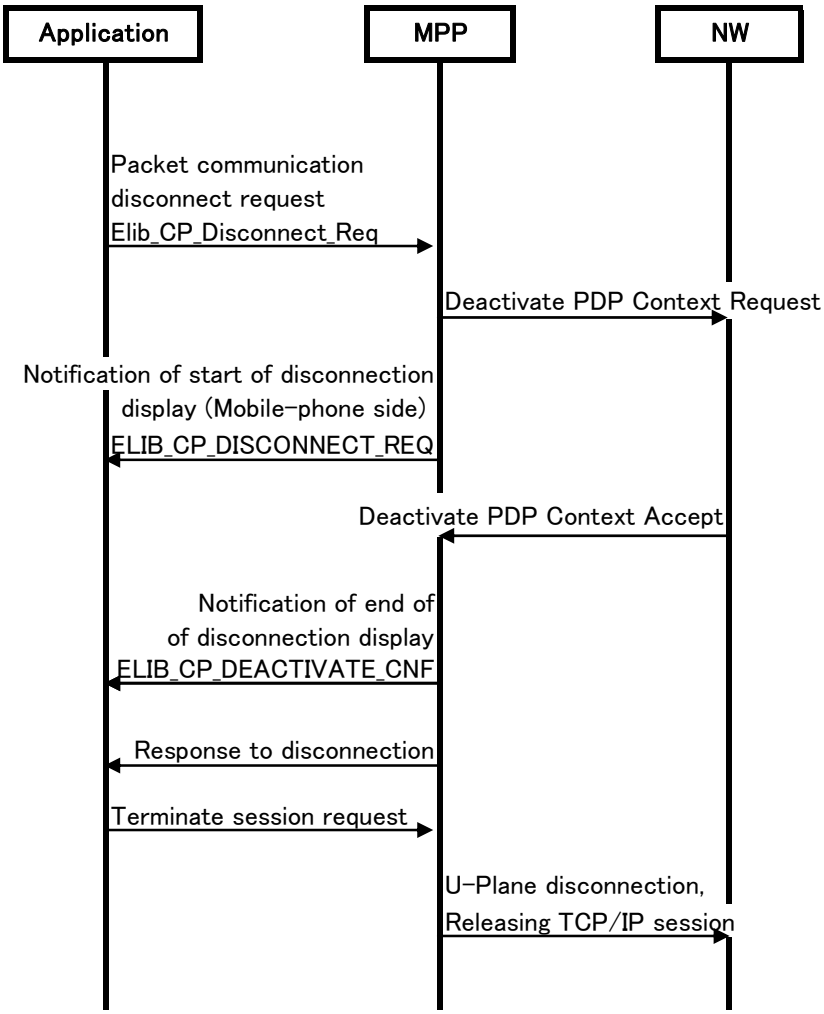
b. Receiving incoming call from NW

Following diagram shows a sample sequence from receiving incoming call from NW side to establish service.



c. Disconnect action from mobile-phone side

Following diagram shows a sample sequence to disconnect from mobile-phone side in communication.



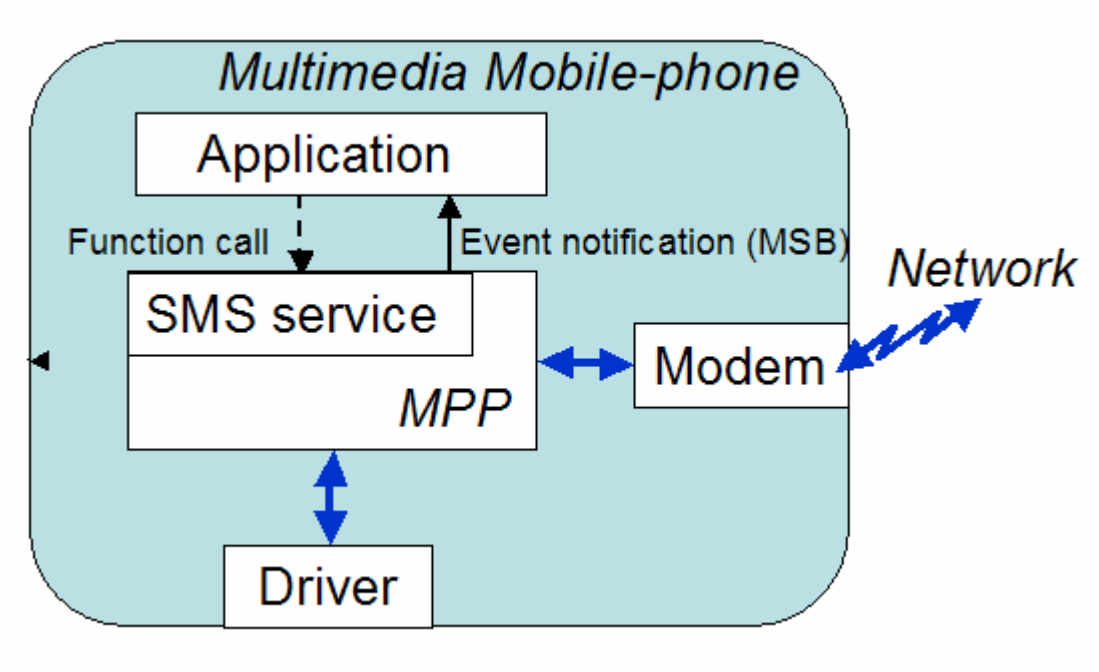
3.1.3 Short message service

3.1.3.1 Short message service

Short message service is a service using SMS protocol. MPP provides SMS service with following functionalities;

- send/receive short messages (SMS)
- retrieve contents of stored information at center side.

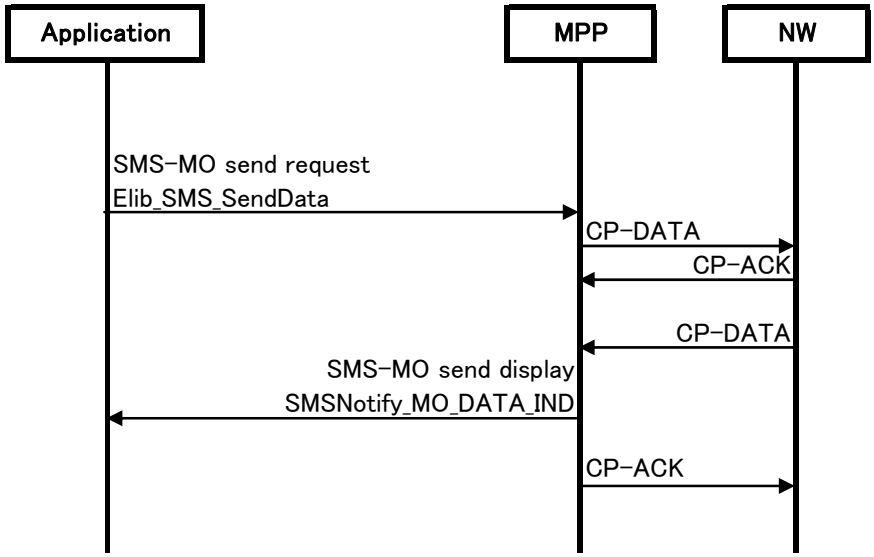
(1)Block diagram



(2)Sequence

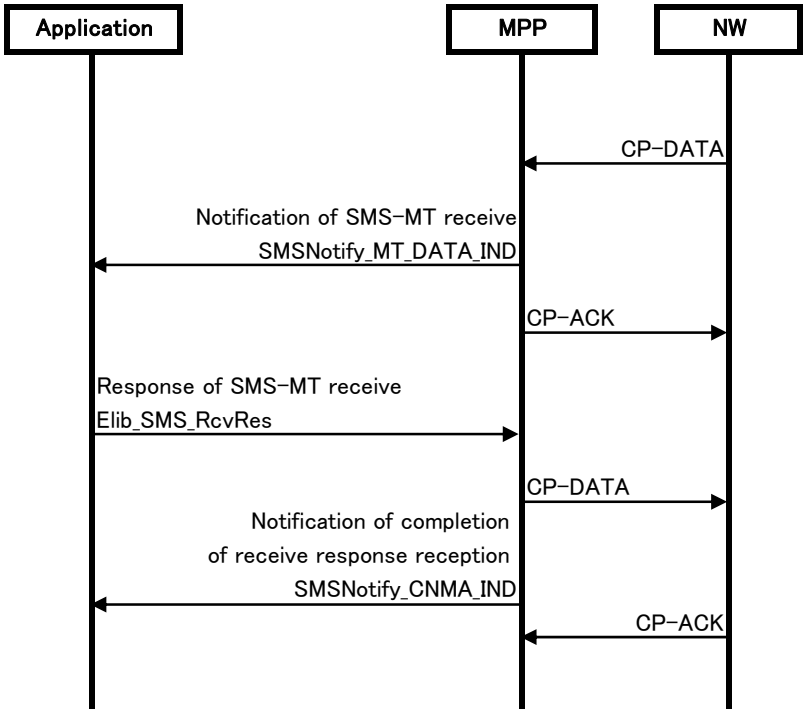
a. SMS transmission

Following diagram shows a sample sequence for SMS transmission action from AP.



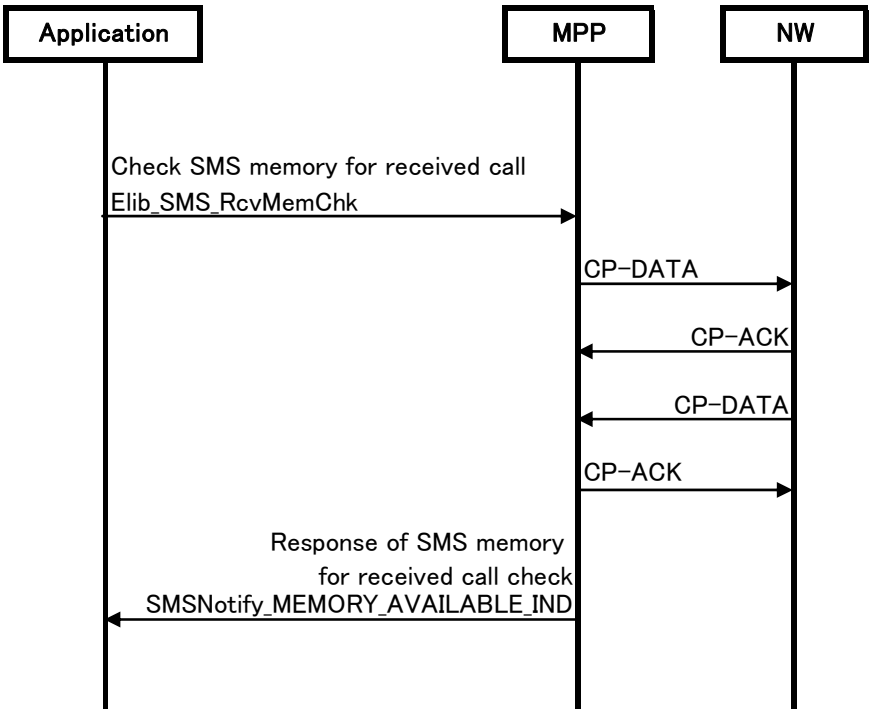
b. SMS receive

Following diagram shows a sample sequence when AP receives SMS incoming call.



c. Retrieve information stored in the SMS center

Following diagram shows a sample sequence for SMS AP to inquire to center and to retrieve information stored in center.



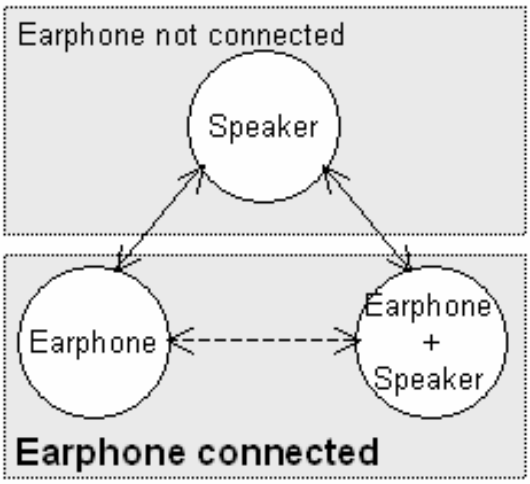
3.1.4 Equipment Service

3.1.4.1 Earphone mode

There are following two modes in earphone mode;

- Earphone + speaker： Both earphone and speaker sound
- Earphone： Only earphone sounds

Mobile phone has following status.



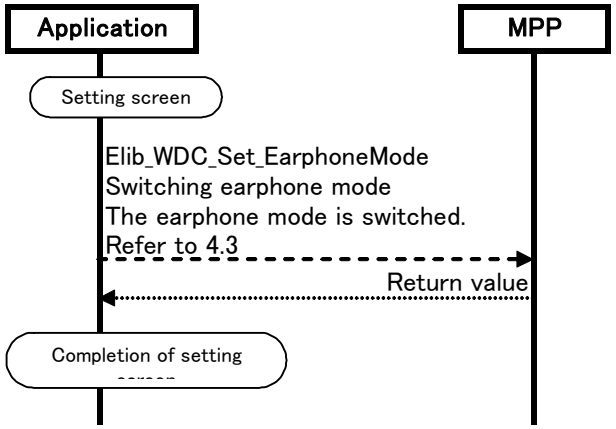
MPP provide following functionalities;

- Switch earphone mode
- Refer the status of earphone connection (connected or not connected)
- Refer earphone mode
- Monitor the change of earphone connection status

(1) Calling Sequence

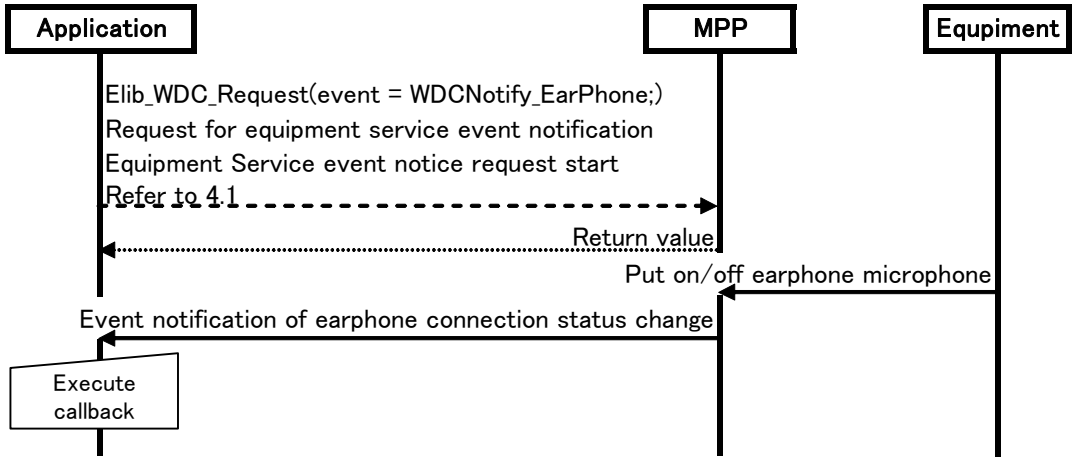
a. Synchronous type

Switching earphone mode, referring earphone connection status and referring earphone mode are synchronous type of APIs. Following diagram shows the sequence of switching earphone mode as typical example.



b. Monitoring the change of earphone connection status

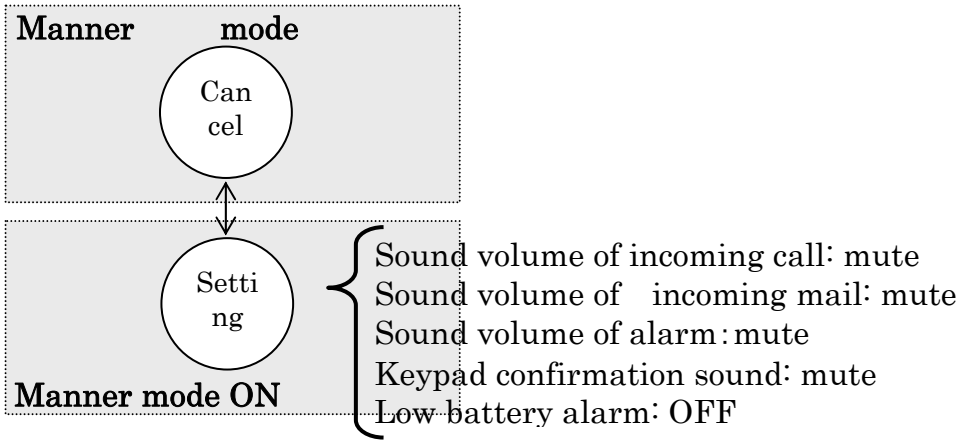
By calling API beforehand from application to request the notification of monitoring the change of earphone connection status, the notification can be received as an event when earphone connection status is changed.



3.1.4.2 Manner mode

Manners mode is the function to mute sound which comes out of a speaker, such as a ring tone and keypad sound to avoid people around mobile phone user.

Mobile phone has following status.

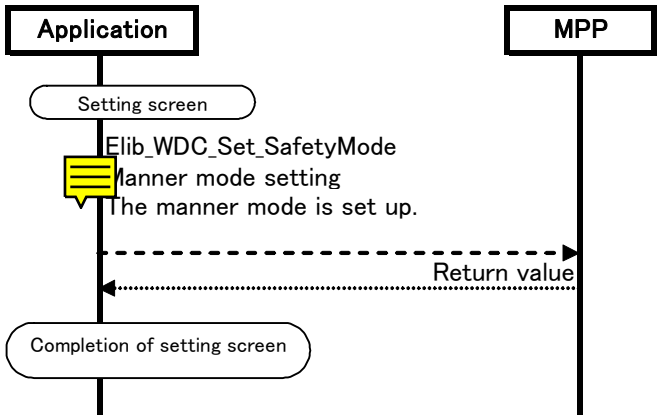


MPP provides following functions.

- Set manner mode ON/OFF
- Monitor manner mode ON/OFF
- Monitoring operation status (internal status) of manner mode

(1) Calling Sequence

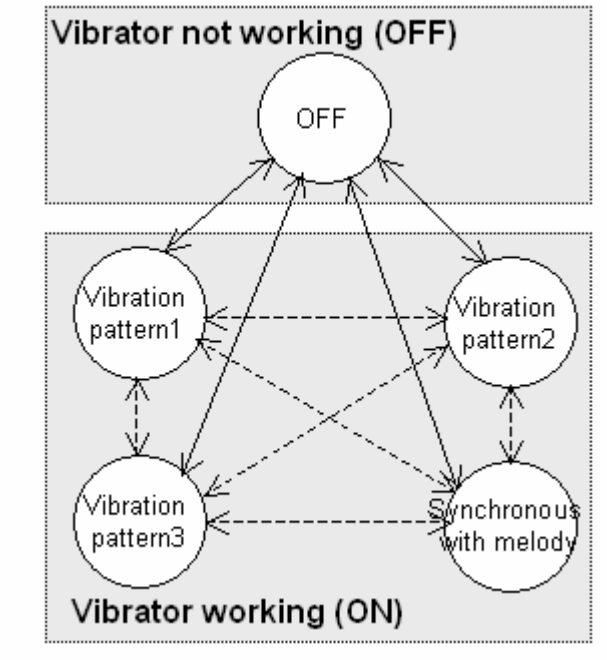
API for manner mode set/reference is of synchronous type. Following diagram shows the sequence of manner mode setting as typical example.



3.1.4.3 Vibrator

Vibrator is a feature to notify receiving incoming call and mail by vibration. Vibrator settings are categorized by receiving incoming call, mail and TV phone, etc. and there are several vibration patterns including synchronized vibration with melody.

Mobile phone has following status.



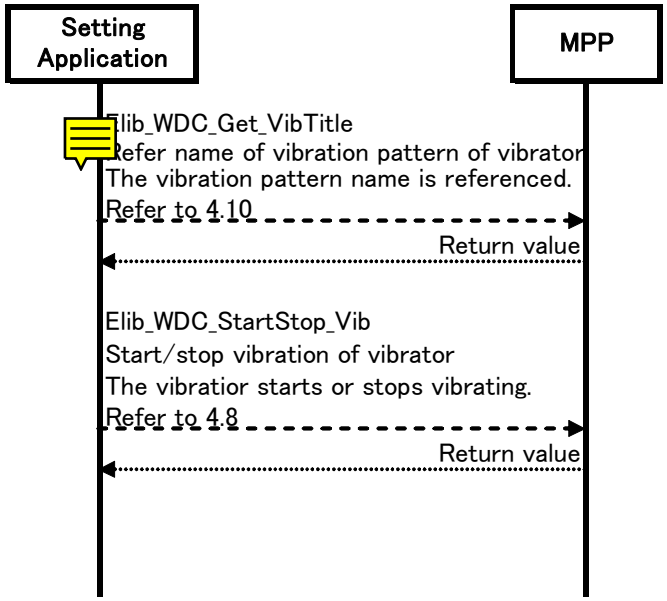
MPP provides following functionalities.

- Start/stop vibration of vibrator.
- Set vibration pattern of vibrator.
- Refer vibration pattern of vibrator.
- Refer current status of vibrator’s operation.

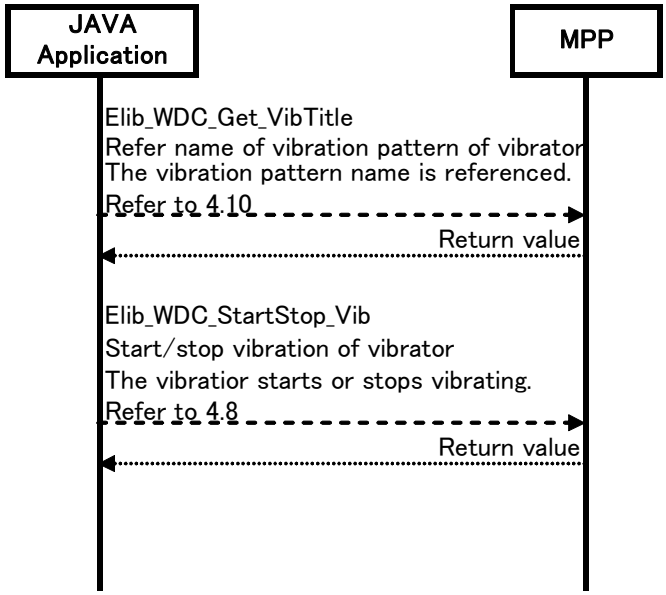
(1) Calling Sequence

API for vibrator set/reference is of synchronous type. Following diagram shows the sequence of starting vibrator as typical example.

a. Setting Application



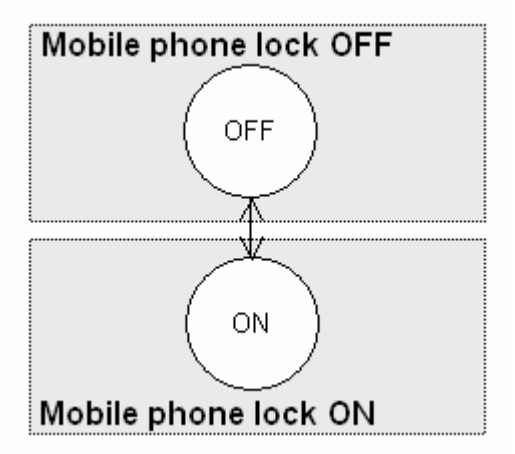
b. JAVA Application



3.1.4.4 Lock of mobile phone

Lock of mobile phone (all lock) is a functionality to disable any operation other than power ON/OFF.

Mobile phone has following status.



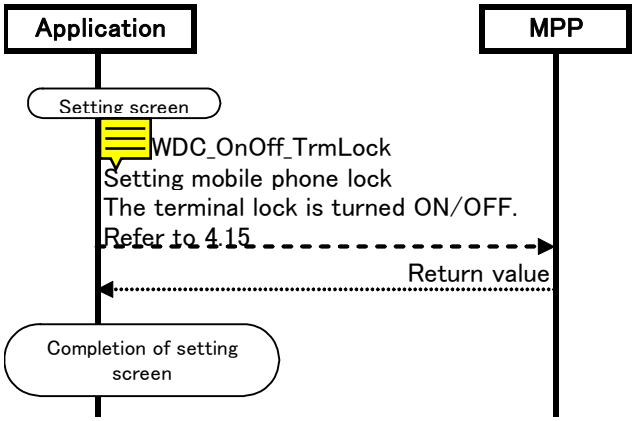
MPP provides following functionalities.

- Set mobile phone lock (all lock) On/OFF.
- Refer mobile phone lock (all lock) status, On/OFF.
- Monitor change of mobile phone lock (all lock) status, On/OFF.

(1) Calling Sequence

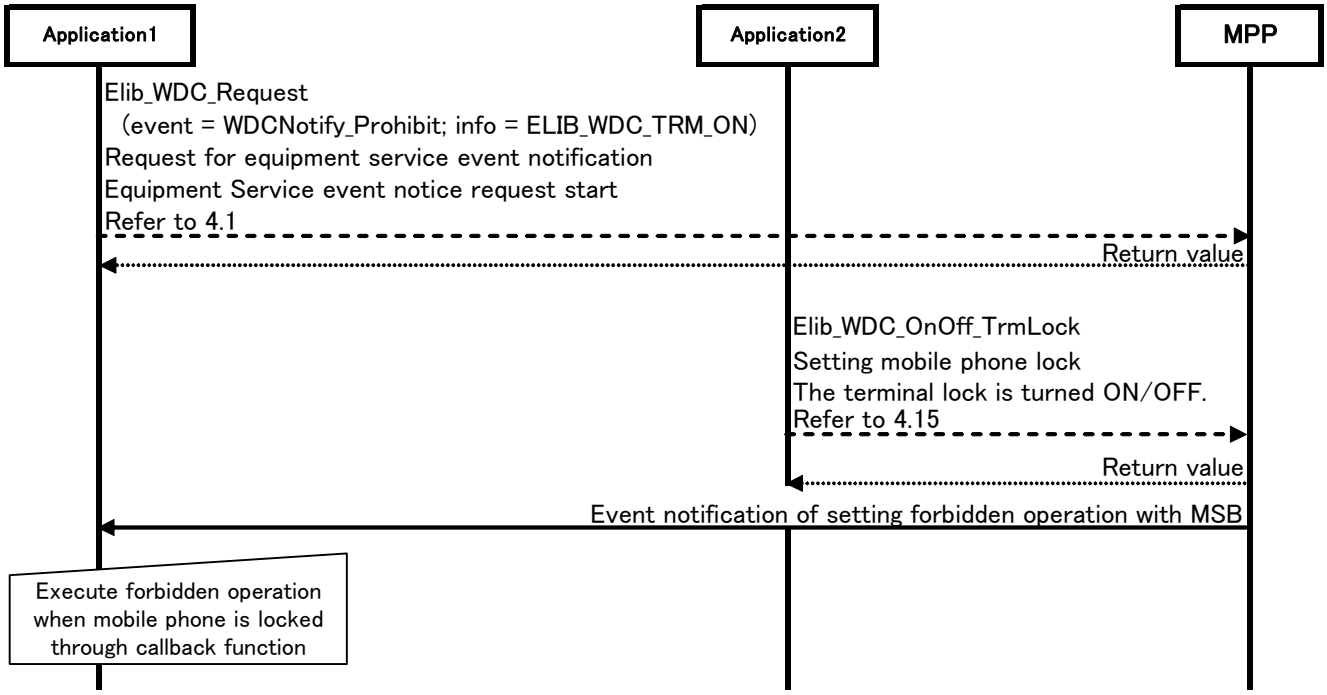
a. Synchronous type

API to set/refer mobile phone lock (all lock) ON/OFF is of synchronous type. Following diagram shows the sequence of setting mobile phone lock (all lock) ON/OFF as typical example.



b. Notification of mobile phone lock ON/OFF request

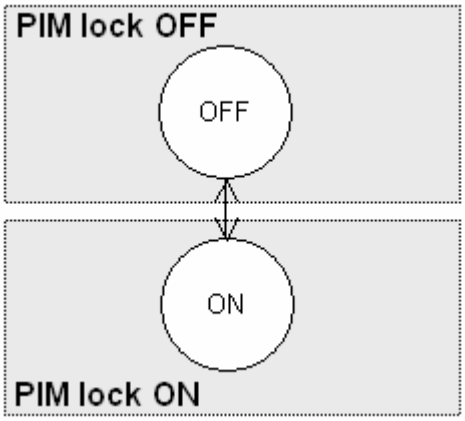
By calling API beforehand from application to request the notification of mobile phone lock ON/OFF, the notification can be received as an event when mobile phone lock (all lock) is set.



3.1.4.5 PIM lock

PIM lock is a functionality to prevent personal information from browsing and rewriting.

Mobile phone has following status.



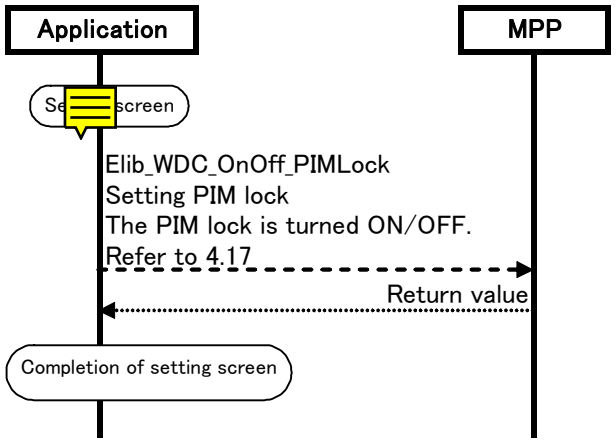
MPP provides following functionalities.

- Set PIM lock ON/OFF.
- Refer PIM lock ON/OFF.
- Monitor the change of PIM lock ON/OFF status.

(1) Calling Sequence

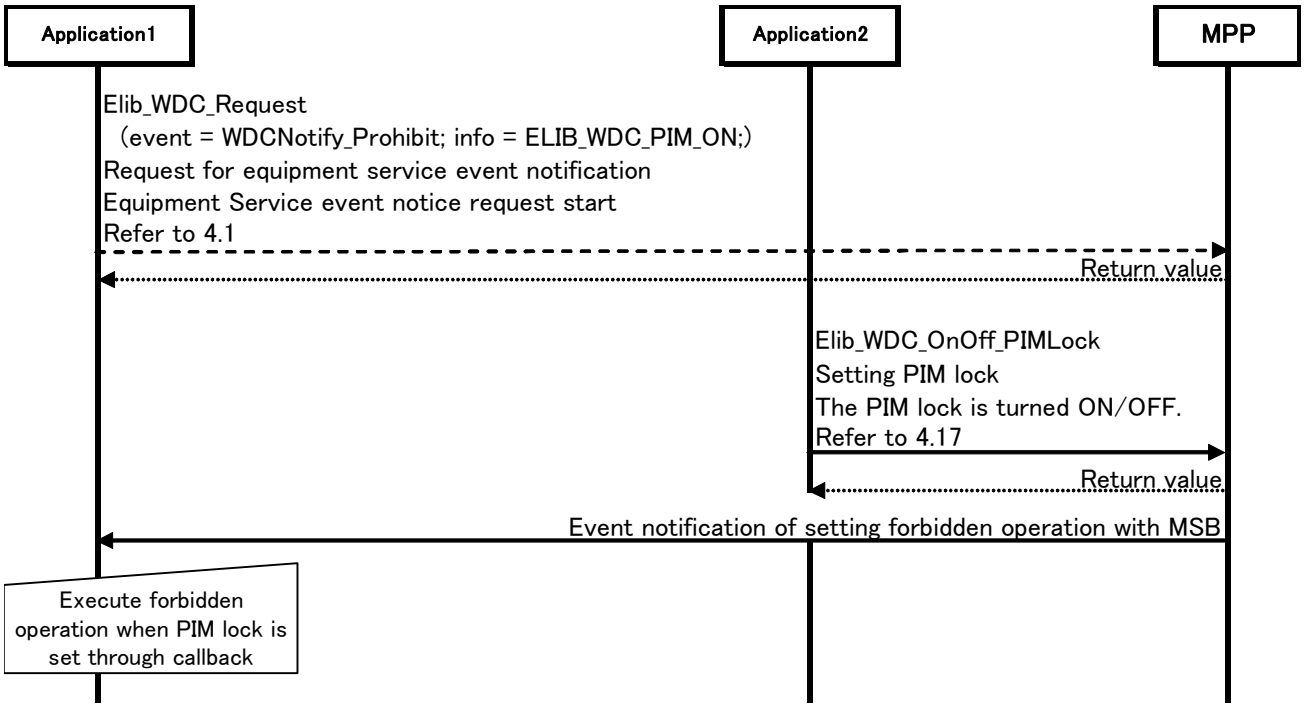
a. Synchronous type

API to set/refer PIM lock ON/OFF is of synchronous type. Following diagram shows the sequence of setting PIM lock ON/OFF as typical example.



b. Notification of PIM lock ON/OFF request

By calling API beforehand from application to request the notification of PIM lock ON/OFF, the notification can be received as an event when PIM lock is set.

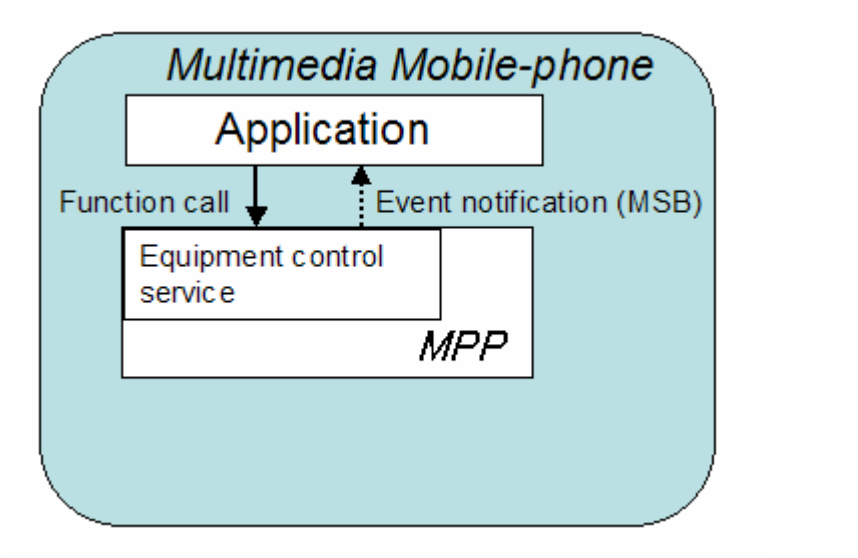


3.1.4.6 Dial call restriction

Dial call restriction is a functionality to restrict dial call other than specified dial number. MPP provides following functionalities.

- Set dial call restriction ON/OFF.
- Monitor the setting of dial call restriction
- Get the status of dial call restriction setting

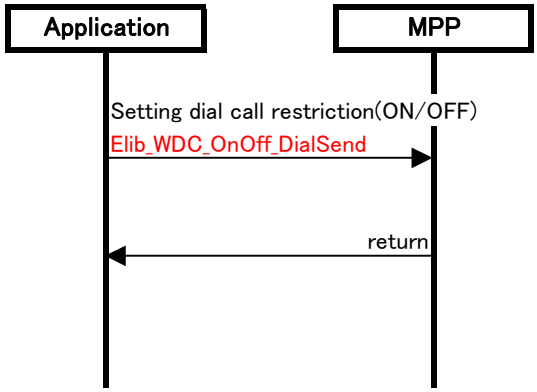
(1)Block diagram



(2)Sequence

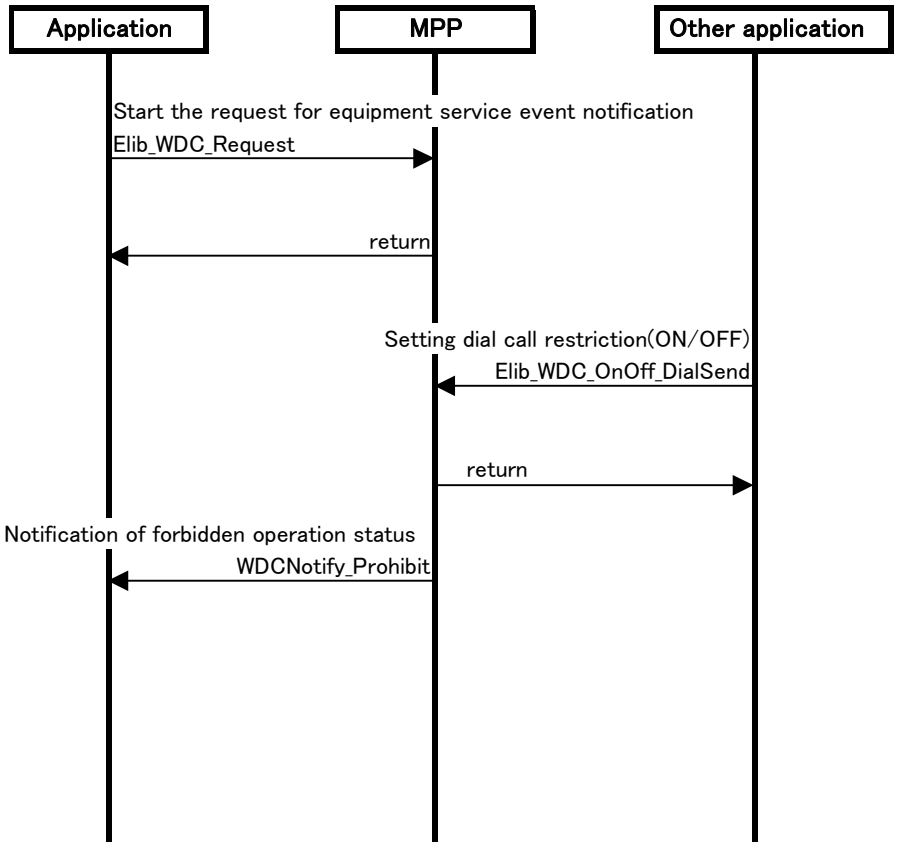
a. Setting dial call restriction ON/OFF

Following diagram shows the sequence of setting dial call restriction ON/OFF.



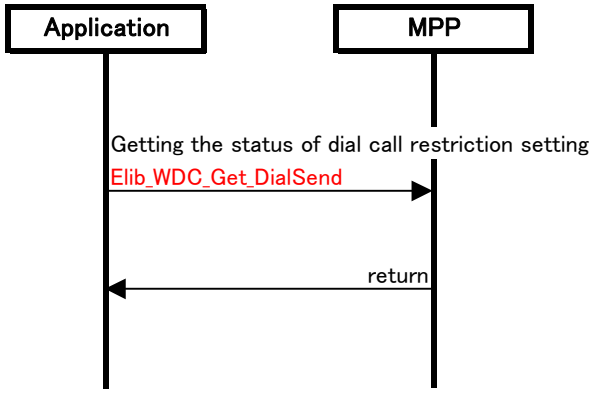
b. Monitoring dial call restriction -setting

Following diagram shows the sample sequence for application to get the notification when the status of dial call restriction setting changes.



c. Getting the status of dial call restriction setting

Following diagram shows the sample sequence for application to refer the status of dial call restriction setting.



(3)Service API to implement functionalities

To create APL, it would be better to use following APIs for equipment services.

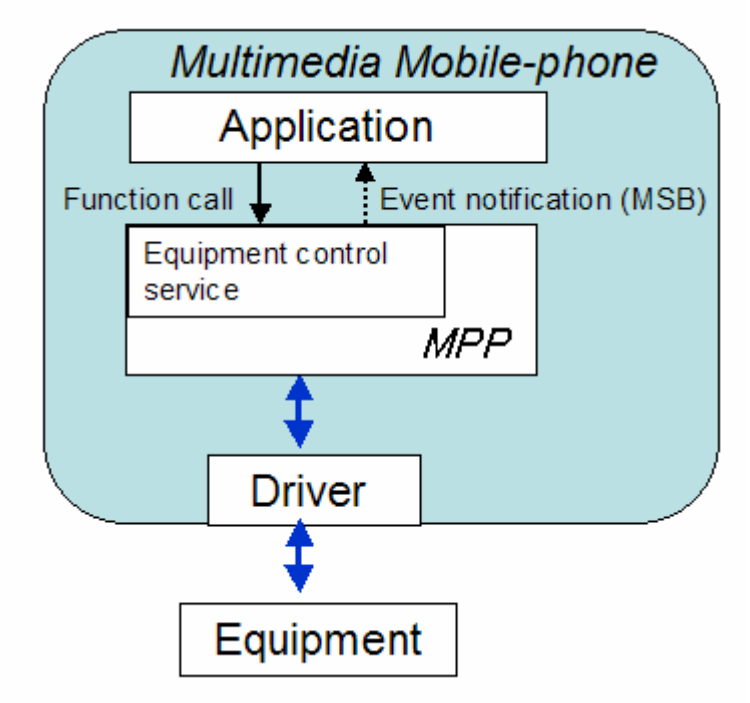
- Elib_WDC_OnOff_DialSend: Setting ON/OFF to forbid dial call
- Elib_WDC_Get_DialSend: Referring ON/OFF to forbid dial call

3.1.4.7 Battery level, pack

MPP provides following functionalities to check battery level and the status of battery pack.

- Referring battery level
- Referring the status of battery pack attached or detached

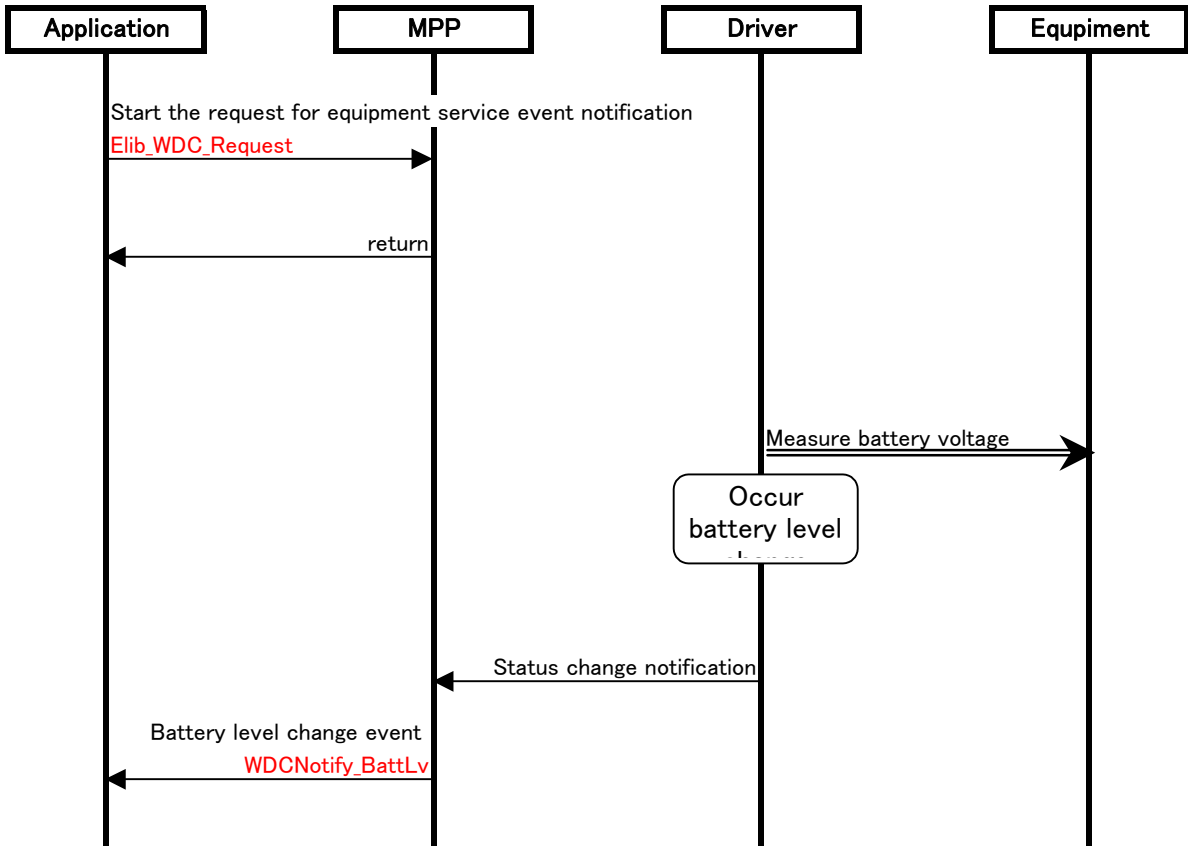
(1)Block diagram



(2)Sequence

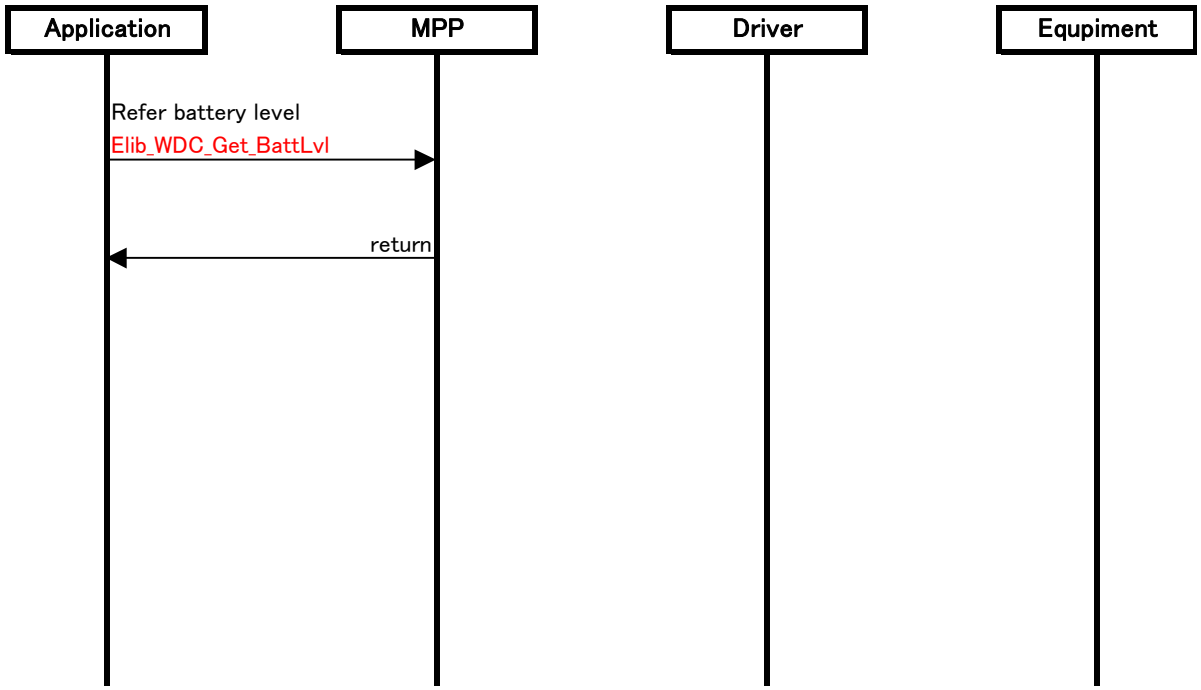
a. Monitoring battery level

Following diagram shows the sample sequence for application to receive event from equipment service. By this setting, application receives notification message when battery level changes.



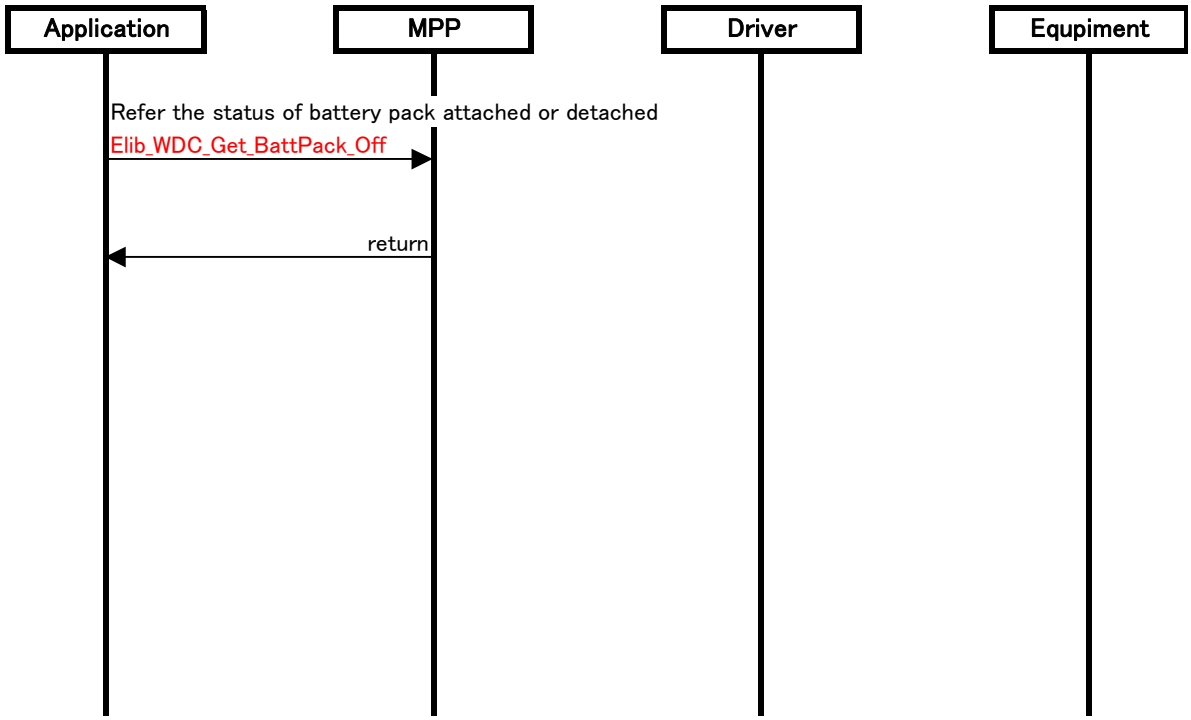
b. Referring battery level

Following diagram shows the sample sequence for application to refer battery level.



c. Refer the status of battery pack attached or detached

Following diagram shows the sample sequence for application to refer the status of battery pack attached or detached.



(3)Service API to implement functionalities

To create APL, it would be better to use following APIs for equipment services.

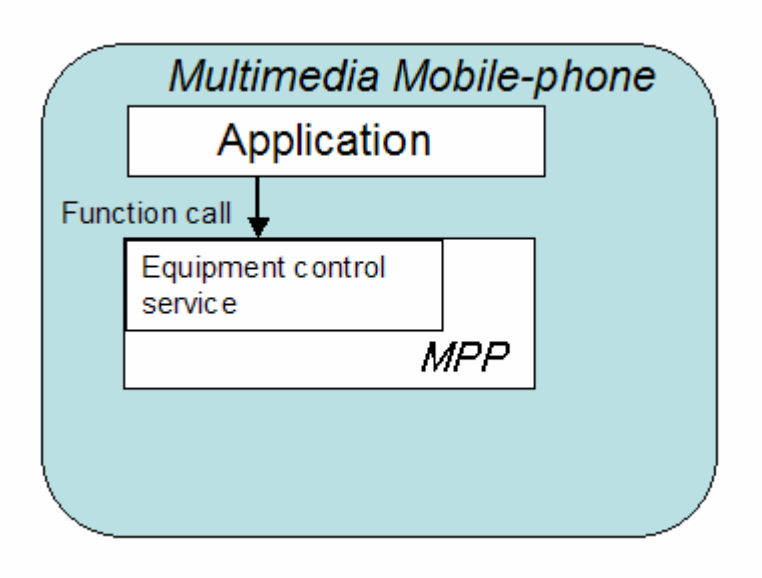
- Elib_WDC_Get_BattLvl: Function to refer the status of battery level
- Elib_WDC_Get_BattPack_Off：Function to refer the status of battery pack attached or detached

3.1.4.8 IMEI reference

IMEI reference is a functionality to refer IMEI stored in the body of mobile phone.
MPP provides following functionality.

- Referring IMEI

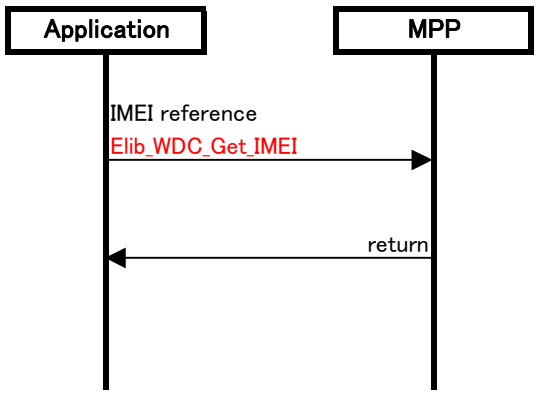
(1)Block diagram



(2)Sequence

a. IMEI reference

Following diagram shows the sample sequence to refer IMEI.



(3)Service API to implement functionalities

To create APL, it would be better to use following APIs for services.

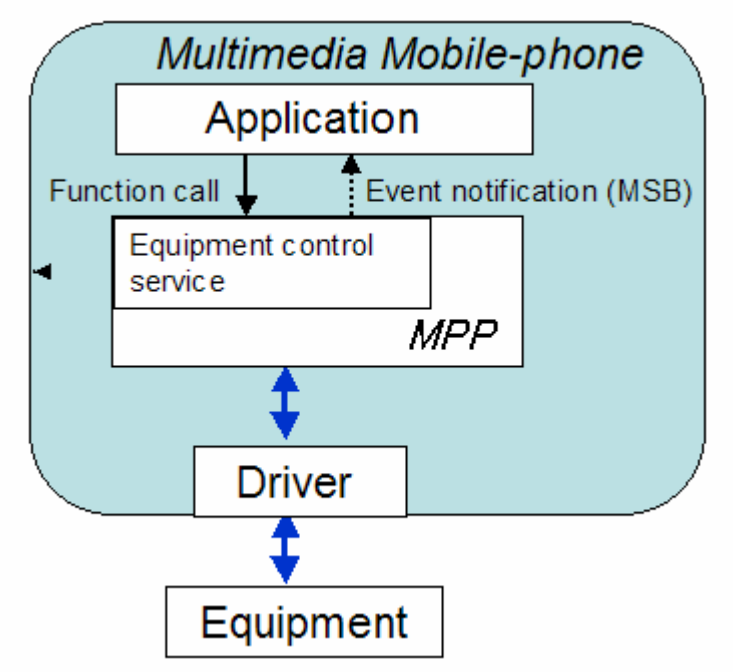
- Elib_WDC_Get_IMEI：IMEI reference

3.1.4.9 USB connection status

USB connection status is a functionality to check the status of USB equipment attached to the body of mobile phone.
MPP provides following functionalities.

- Monitoring USB connection status
- Referring USB connection status

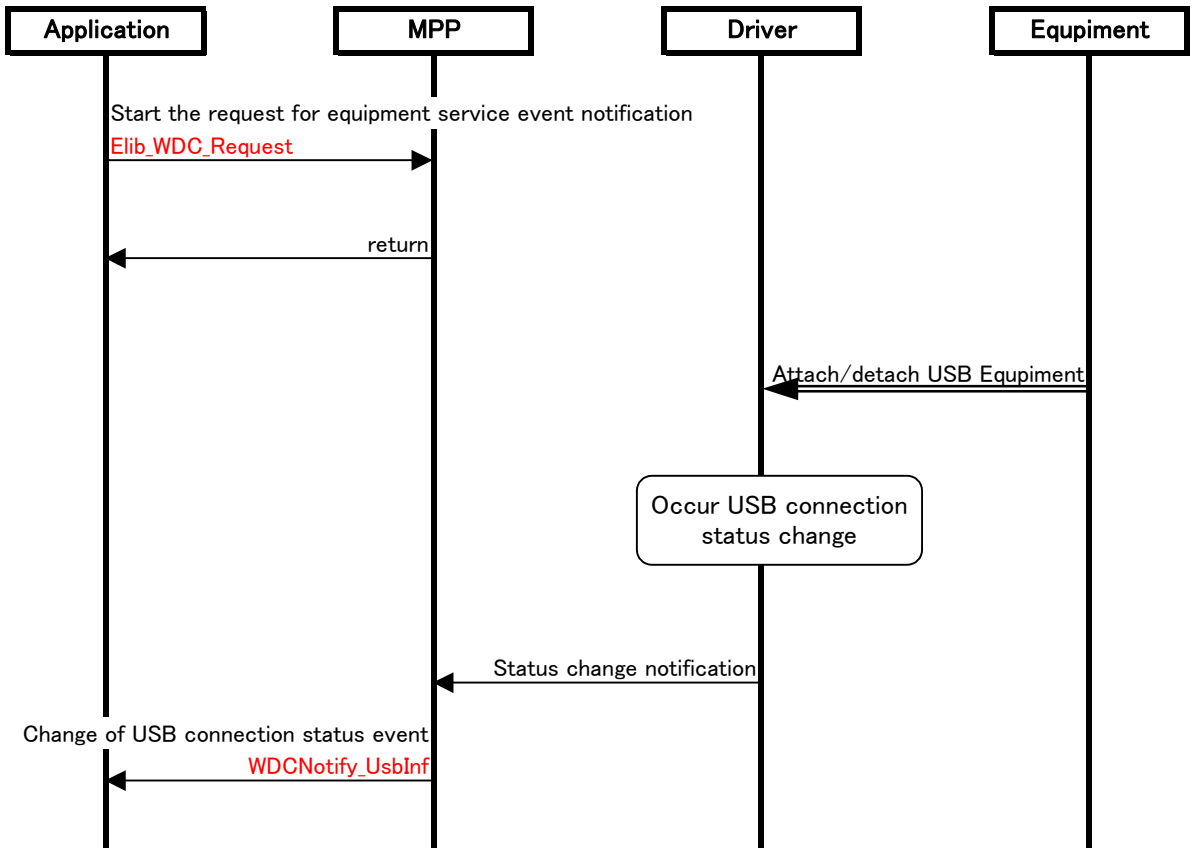
(1)Block diagram



(2)Sequence

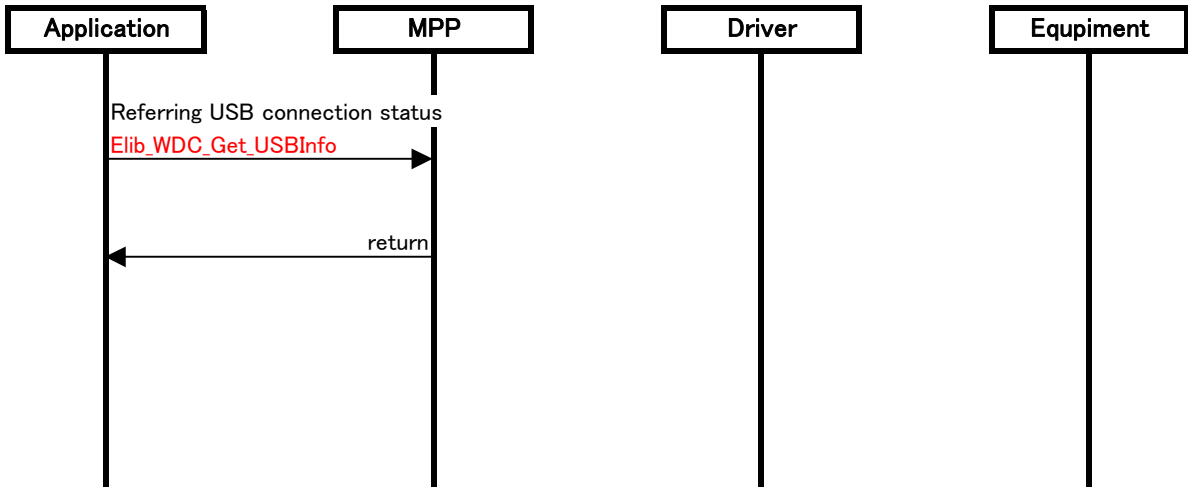
a. Monitoring USB connection status

Following diagram shows the sample sequence for application to receive event from equipment service.



b. Referring USB connection status

Following diagram shows the sample sequence when USB connection status changes.



(3)Service API to implement functionalities

To create APL, it would be better to use following APIs for services.

・Elib_WDC_Get_USBInfo：Referring USB connection status

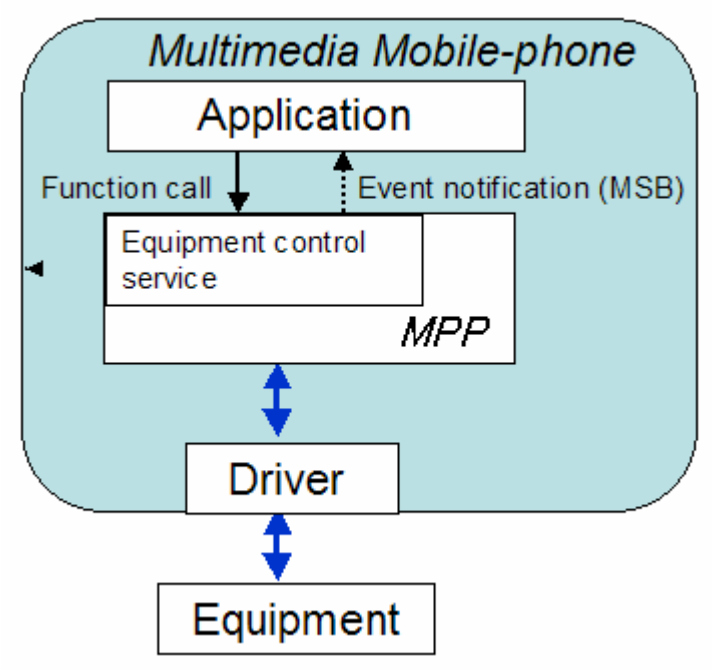
3.1.4.10 Open/close status

Open/close status is a functionality to check status change when equipment is opened and closed.
MPP provides following functionalities.

- Monitoring open/close status
- Referring open/close status

(1)Block diagram

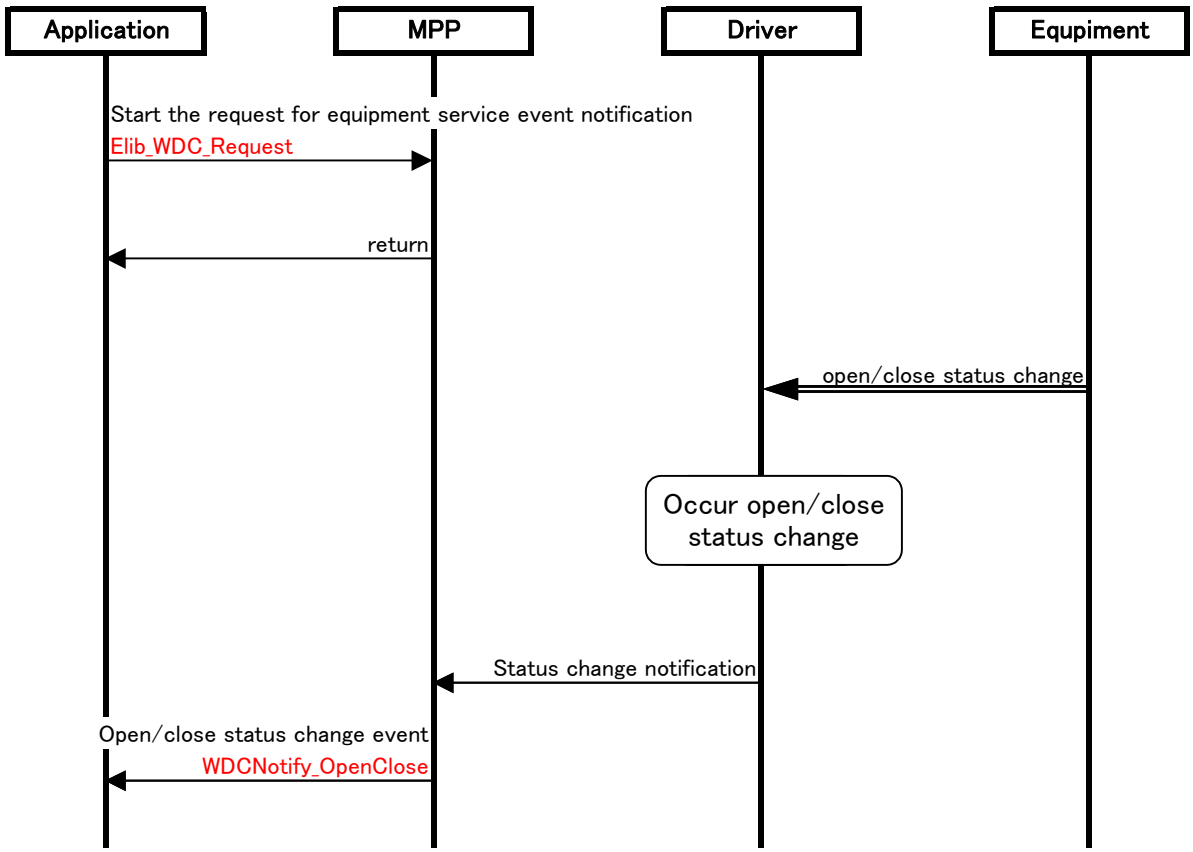
Refer to "Block diagram.ppt"



(2)Sequence

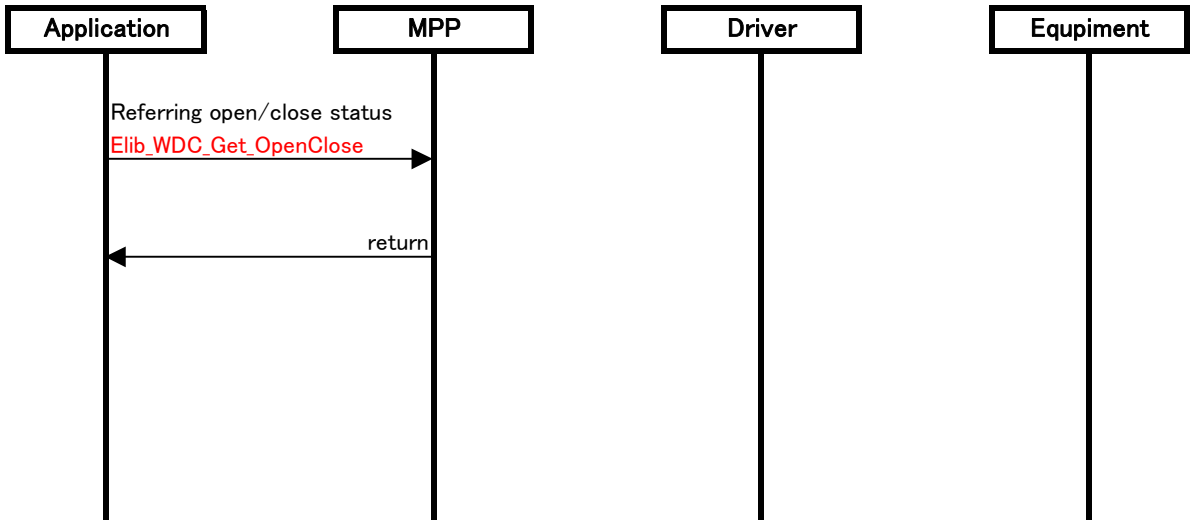
a. Monitoring open/close status

Following diagram shows the sample sequence for application to receive event from equipment service.



b. Referring open/close status

Following diagram shows the sample sequence to refer open/close status of mobile phone.



(3)Service API to implement functionality

To create APL, it would be better to use following APIs for services.

- Elib_WDC_Get_OpenClose：Referring open/close status